Distributed Self Fault Diagnosis in Wireless Sensor Networks using Statistical Methods

Meenakshi Panda

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela, Odisha, 769008, India

March 2015

Distributed Self Fault Diagnosis in Wireless Sensor Networks using Statistical Methods

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Meenakshi Panda
(Roll: 509CS106)

under the guidance of

Prof. Pabitra Mohan Khilar

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India
March 2015

dedicated to my parents with love...

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India.

Dr. Pabitra Mohan Khilar
Assistant Professor

December 22, 2014

Certificate

This is to certify that the work in the thesis entitled ***Distributed Self Fault Diagnosis in Wireless Sensor Networks using Statistical Methods*** by ***Meenakshi Panda*** is a record of an original research work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Doctor in Philosophy in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Pabitra Mohan Khilar

Acknowledgment

"The will of God will never take you where Grace of God will not protect you."
Thank you God for showing me the path...
I owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, I would like to express my sincere gratitude to my advisor, Prof. Pabitra Mohan Khilar for providing me with a platform to work on challenging areas of distributed self fault diagnosis in Wireless Sensor Networks. His profound insights and attention to details have been true inspirations to my research.

I am thankful to *Prof. S. K. Jena*, *Prof. S. K. Rath*, *Prof. B. Majhi*, *Prof. R. Baliarsingh*, *Prof. D. P. Mohapatra*, *Prof. A. K. Turuk*, and *Prof. B. D. Sahoo* of Computer Science and Engineering Department and Prof. K. B. Mohanty of Electrical Engineering Department for extending their valuable suggestions and help whenever I approached them.

It is my great pleasure to show indebtedness to my friends for their help during the course of this work. I acknowledge all staff, research scholars and juniors of CSE Department, NIT Rourkela for helping me during my research work. I am grateful to NIT Rourkela for providing me adequate infrastructure to carry out the present investigations.

I take this opportunity to express my regards and obligation to my family members whose support and encouragement I can never forget in my life. I wish to appreciate and thank my daughter, Priyanshi, for allowing me to stay away in department for writing this thesis. Without the constant support and encouragement of my husband, Trilochan, I could hardly have completed this work. His unending patience, encouragement and understanding had made it all possible, and meaningful. This thesis is for him.

Meenakshi Panda

Abstract

Wireless sensor networks (WSNs) are widely used in various real life applications where the sensor nodes are randomly deployed in hostile, human inaccessible and adversarial environments. One major research focus in wireless sensor networks in the past decades has been to diagnose the sensor nodes to identify their fault status. This helps to provide continuous service of the network despite the occurrence of failure due to environmental conditions. Some of the burning issues related to fault diagnosis in wireless sensor networks have been addressed in this thesis mainly focusing on improvement of diagnostic accuracy, reduction of communication overhead and latency, and robustness to erroneous data by using statistical methods. All the proposed algorithms are evaluated analytically and implemented in standard network simulator NS3 (version 3.19).

A distributed self fault diagnosis algorithm using neighbor coordination (DSFDNC) is proposed to identify both hard and soft faulty sensor nodes in wireless sensor networks. The algorithm is distributed (runs in each sensor node), self diagnosable (each node identifies its fault status) and can diagnose the most common faults like stuck at zero, stuck at one, random data and hard faults. In this algorithm, each sensor node gathered the observed data from the neighbors and computes the mean to check the presence of faulty sensor node. If a node diagnoses a faulty sensor node in the neighbors, then it compares observed data with the data of the neighbors and predicts its probable fault status. The final fault status is determined by diffusing the fault information obtained from the neighbors. The accuracy and completeness of the algorithm are verified based on the statistical analysis over sensors data. The performance parameters such as diagnosis accuracy, false alarm rate, false positive rate, total number of message exchanges, energy consumption, network life time, and diagnosis latency of the DSFDNC algorithm are determined for different fault probabilities and average degrees and compared with existing distributed fault diagnosis algorithms.

To enhance the diagnosis accuracy, another self fault diagnosis algorithm is proposed based on hypothesis testing (DSFDHT) using the neighbor coordination approach. The Newman-Pearson hypothesis test is used to diagnose the soft fault status of each sensor node along with the neighbors. The algorithm can diagnose the faulty sensor node when the average degree of the network is less. The diagnosis accuracy, false alarm rate and false positive rate performance of the DSFDHT algorithm are improved over DSFDNC for sparse wireless sensor networks by keeping other performance parameters nearly same.

The classical methods for fault finding using mean, median, majority voting and hypothesis testing are not suitable for large scale wireless sensor networks due to large deviation in transmitted data by faulty sensor nodes. Therefore, a modified three sigma edit test based self fault diagnosis algorithm (DSFD3SET) is proposed which diagnoses in an efficient manner over a large scale wireless sensor networks. The diagnosis accuracy, false alarm rate, and false positive rate of the proposed algorithm improve as compared to that of the DSFDNC and DSFDHT algorithms. The algorithm enhances the total number of message exchanges, energy consumption, network life time, and diagnosis latency, because the proposed algorithm needs less number of message exchanges over the algorithms such as DSFDNC and DSFDHT.

In the DSFDNC, DSFDHT and DSFD3SET algorithms, the faulty sensor nodes are considered as soft faulty nodes which behave permanently. However in wireless sensor networks, the sensor nodes behave either fault free or faulty during different periods of time and are considered as intermittent faulty sensor nodes. Diagnosing intermittent faulty sensor nodes in wireless sensor networks is a challenging problem, because of inconsistent result patterns generated by the sensor nodes. The traditional distributed fault diagnosis (DIFD) algorithms consume more message exchanges to obtain the global fault status of the network. To optimize the number of message exchanges over the network, a self fault diagnosis algorithm is proposed here, which repeatedly conducts the self fault diagnosis procedure based on the modified three sigma edit test over a duration to identify the intermittent faulty sensor nodes. The algorithm needs less number of iterations to identify the intermittent faulty sensor nodes. The simulation results show that, the performance of the DHISFD3SET algorithm improves in diagnosis accuracy, false alarm rate and false positive rate over the DIFD algorithm.
***Keywords*:** Wireless Sensor Networks, Hard and Soft fault, Intermittent Fault, Hypothesis Testing, Three Sigma Edit Test, Normal Distribution, Distributed Self Fault Diagnosis.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ANN	Artificial Neural Network
BC	Broadcast comparison-based model
BGM	Barsi, Grandoni, and Maestrini test-based model
BPNN	Back Propagation Neural Network
CI	Confidence Interval
CSFD	Collaborative Sensor Fault Diagnosis
DFD	Distributed Fault Diagnosis
DHISFD3SET	Distributed Hard Intermittent Self Fault Diagnosis using Three Sigma Edit Test
DIFD	Distributed Intermittent Fault Diagnosis
DSFD	Distributed Self Fault Diagnosis
DSFDHT	Distributed Self Fault Diagnosis using Hypothesis Testing
DSFDNC	Distributed Self Fault Diagnosis using Neighbor co-ordination approach
DSFD3SET	Distributed Self Fault Diagnosis using Three Sigma Edit Test
FAR	False Alarm Rate
FPR	False Positive Rate
GC	Generalized comparison-based model
GDC	generalized distributed comparison-based model
HK	Hakimi and Kreutzer test-based model
IDFD	Improved Distributed Fault Diagnosis
LLSE	Linear Least Square Estimation
MANET	Mobile Adhoc Network
MM	Miroslaw Malek comparison-based model
MM*	Chwa and Hakimi comparison-based model
MP	Neyman Pearson Most Powerful test
NP	Neyman Pearson hypothesis testing method
PMC	Preparata, Metze, and Chien test-based model
PNN	Probablistic Neural Network
RNN	Recurrent Neural Network
ROI	Regions Of Interest
RSSI	Received Signal Strength Indication
WSNs	Wireless Sensor Networks

Chapter 1

Introduction

Chapter 1

Introduction

We interact with the physical world through our eyes, ears, nose, mouth, hands, and of course, our brain. In addition, we create instruments to augment our capabilities. With the advance in computing, communication, and microelectronic mechanical system technologies, we are getting closer to the physical world and monitoring and managing it. The wireless sensor networks (WSNs) open a door for potential real world applications. A sensor network is a distributed system, consisting of thousands of physically embedded, unattended, and often, untethered devices. WSNs are more prone to errors due to various unavoidable circumstances of natural calamities. Therefore, efficient fault diagnosis in WSNs is necessary to maintain the quality of service of WSNs.

1.1 Introduction

In recent years, wireless sensor networks (WSNs) have gained worldwide scientific interest due to their ease of deployment and wide range of applications starting from terrestrial to underwater scenarios [1]. WSNs are equipped with tiny, inexpensive and intelligent sensor nodes. It is an infra-structureless network and runs with resource constraints such as limited battery power, short communication range, low bandwidth, and limited processing and storage on each sensor node. In recent past, WSNs impact in our daily life due to their services such as remote environmental monitoring, source localization, target tracking, event detection, security, event boundary detection, and target localization [2].

Sensor nodes used in various application domains are expected to operate autonomously as they are deployed in unattended and hostile environments. Due to this, the sensor nodes are prone to have faults. The root cause of sensor fault is system disorder which occurs due to the mechanical or electrical problems in internal circuits of the sensor node, environmental degradation, battery depletion, or hostile tampering, *etc.* The sensor faults are broadly categorized into two types such as crash faults where a sensor node becomes inactive in the network and soft fault where the sensor node behaves arbitrarily [3]. The sensor fault may occur due to the failure of a component such as microprocessor, transceiver, memory subsystems, energy source, sensors, and actuators or environmental noise. As faults are inevitable in WSNs, it is crucial to determine the set of fault free and faulty sensor nodes. The process of identifying both fault free and faulty sensor nodes in a wireless sensor networks is known as distributed sensor network diagnosis which is the main focus of this research work.

In order to reduce the communication and computation overhead in WSNs, one of the best alternative diagnosis algorithm is the self fault diagnosis algorithm for WSNs [4, 5]. In self fault diagnosis approach, every sensor node identifies its fault status based on the observed data in its neighborhood instead of the observed data from all the sensor nodes in WSNs unlike in distributed diagnosis [6]. Therefore, neighbor coordination is an important methodology to improve the communication and computation overhead in sensor networks, which is our main interest in this dissertation.

It is also necessary to investigate the most frequently occurred faults in different components of WSNs with an aim to propose communication, computation and memory efficient self fault diagnosis algorithm. The self fault diagnosis algorithms need to be evaluated by using generic parameters such as diagnosis accuracy, false alarm rate, false positive rate, diagnosis latency, message exchange, energy consumption, and network life time [7]. The performance of the self fault diagnosis algorithm depends on the statistics of the observed data in sensor node vicinity. The statistical methods such as neighbor coordination, hypothesis testing, three sigma edit test, and modified three sigma edit test are considered here to improve the performance of the self fault diagnosis algorithm.

The rest of this chapter is organized as follows. A brief description of fault, error and failure of sensor nodes is presented in Subsection 1.1.1. Classification, causes, errors, and sources of faults are discussed in Subsection 1.1.2. The fault management is discussed in Subsection 1.1.3. Definitions and terminologies used for measuring the performance of the proposed algorithms are presented in Subsection 1.1.4. Section 1.2 presents the motivation of the proposed works. The objective of the research is given in Section 1.3 and finally, the major contribution and organization of the thesis are discussed in Section 1.4 and 1.5 respectively.

1.1.1 Faults, Errors and Failures of Sensor nodes in WSNs

The fault, error and failure are the three important, interrelated, and generic words used in the area of fault diagnosis [8]. The unexpected behavior of the sensor node is popularly known as sensor fault. When the faults occur in a sensor node, it either does not report to the surrounding sensor nodes (hard fault) [9], or reports with erroneous data (soft fault) [10], or reports with uncertain data (soft fault), report sometimes with fault free information and sometimes with faulty information (intermittent fault) [11]. The presence of a sensor error does not mean that a sensor is hard faulty due to the fact that sometimes it produces erroneous data because of the environmental noise or malicious activities which are known as soft faulty sensor nodes. The presence of sensor fault will always lead to the sensor error [12].

A fault in the sensor node causes a sensor error which in turn causes sensor failure. In other words, the cause of the sensor failure is an error reported by sensor node and causes of error in sensor node is the occurrence of faults in sensor nodes. The failure in sensor nodes leads to another part of the current wireless sensor network or to another WSN on which the operation of current WSN depends. The relation among sensor fault, error, and failure are depicted in Figure 1.1.

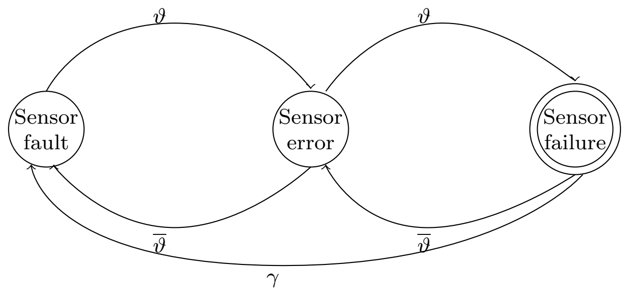


Figure 1.1: An automaton for illustrating relationship among sensor fault, error and failure

1.1.2 Fault classification

Sensor faults are classified into various categories [13–15] which are summarized in Figure 1.3. Based on the behavior of failed nodes or links, the sensor faults are classified into two categories namely hard and soft faults. A sensor node or link suffering with hard faults is unable to communicate with other sensor nodes where as a soft faulty sensor node or link continue to participate in normal operation of the network with altered behaviors. Similarly, due to the persistence of fault, the sensor faults can be classified into two categories namely permanent and temporary fault. Permanent faults are hardware or software faults due to which the sensor node remains silent throughout the life span of the network [14]. However, the temporary faults allow the networking components to actively participate in the operation of the network. Based on the duration, a sensor node or link remains permanent or temporary faulty. This fault is also known as hard or crash fault, which is used interchangeably throughout the thesis.

Temporary fault is furthered classified into three categories such as transient, intermittent and Byzantine fault. A transient fault lasts for a small duration which is called a spike and allows the network to be functional for the remaining time. An intermittently faulty component behaves sometimes faulty and in other time fault free during the life of WSNs. A Byzantine faulty component can be arbitrary faulty includes any types of faults, therefore, is a challenging task to detect and diagnose.

Another way of classifying sensor faults based on underlying causes are presented by Barborak *et al.* [14], where sensor faults are classified as: fail-stop, crash, omission, timing, incorrect computation and Byzantine fault. An order of occurrence of these types of faults is depicted in Figure 1.4. Fail-stop and crash faults are hard or permanent faults, and all others can be considered as soft faults.

The fault that occurs when a sensor component ceases operation due to depletion of battery and alerts to its one-hop neighbors is known as fail-stop fault. A sensor component suffering with crash fault remains silent in WSNs till its replacement by an external agent. Omission faulty sensor components do not respond to the sink node at the right time and also fails to send the desired information to the sink node on time or fails to relay the received message to its neighbors on time. Like the omission fault, the sensor components suffering with timing fault work normally, but transmit or receive the correct data either too soon, or too late. When a sensor component is suffering with incorrect computation fault, it fails to send the actual sensed data or processed information to other network nodes even though the sensing element of the sensor node perceived with the actual data. Similar to an incorrect computation fault, a Byzantine faulty component also gives arbitrary value at different time instants. All the above said soft faults may be a natural or human-made fault and can be either intermittent or transient in nature. The detail description of the nature of all fault types is summarized in Figure 1.2.

Based on the voltage supplied to the sensor node, the sensor node suffers from another type of fault called spike fault in which a voltage spike (or impulse) is superimposed on the sensor measurement which generates arbitrary value. This type of fault may be transient or intermittent or permanent in nature [16, 17].

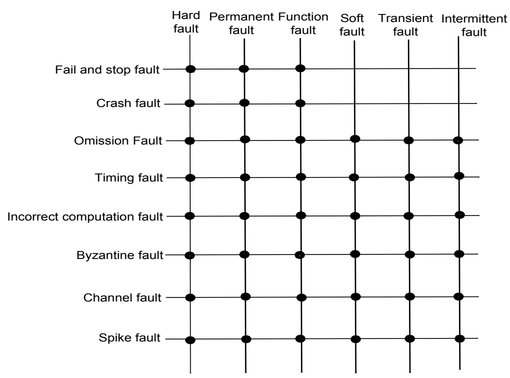


Figure 1.2: Detail description of fault types used in WSNs

Developing self fault diagnosis algorithms for diagnosing each and every fault in sensor nodes and links is not only challenging, but also not feasible for energy constraint battery operated WSNs. In order to address the most frequently occurred faults in WSNs, we have proposed the self fault diagnosis algorithm considering the hard and soft faults. In soft fault, erroneous data due to sensor node's incorrect computation and intermittently faulty sensor nodes are considered. Only sensor node faults are considered assuming that links are fault free which are usually taken care by underlying communication network protocol (for example 802.15.4).

Based on the data generated by the faulty sensor components, the soft fault is again partitioned into three sub categories, namely constant, noise and short fault [18]. In constant fault, each of the soft faulty sensor nodes generates constant value which is either too large or small compared to a normal reading of the sensor component. When a significant change occurs between any two successive readings of the sensor nodes the faults are categorized as short faults. Similarly, in noise fault, the variance of the sensor reading increases. All the above said fault types can be detected by the sensor node itself without any other neighboring sensor nodes reading. The faults are identified based on the observed data by each and every sensor node in wireless sensor networks. It depends on its own data and performs the computation based on the observed data. It does not need any communication to other sensor nodes except the neighboring nodes thereby reducing the communication overhead. In fact, this lead to less energy overhead as energy consumption is directly proportional to number of messages communicated.

Causes/ Sources of the Sensor Fault

The key sources of sensor failure are due to damage of the transceiver or any internal circuit of the sensor node due to the natural calamities, calibration error, malfunctioning hardware, hostile environment, low battery and link failure [8]. Though the calibration during deployment is performed, sensors throughout their deployed lifetime may drift. This in turn lowers the accuracy of sensor measurements. Three different types of calibration errors are reported by Ni *et al.* [19] namely offset faults (sensor measurements offset from the ground truth by a constant amount), gain

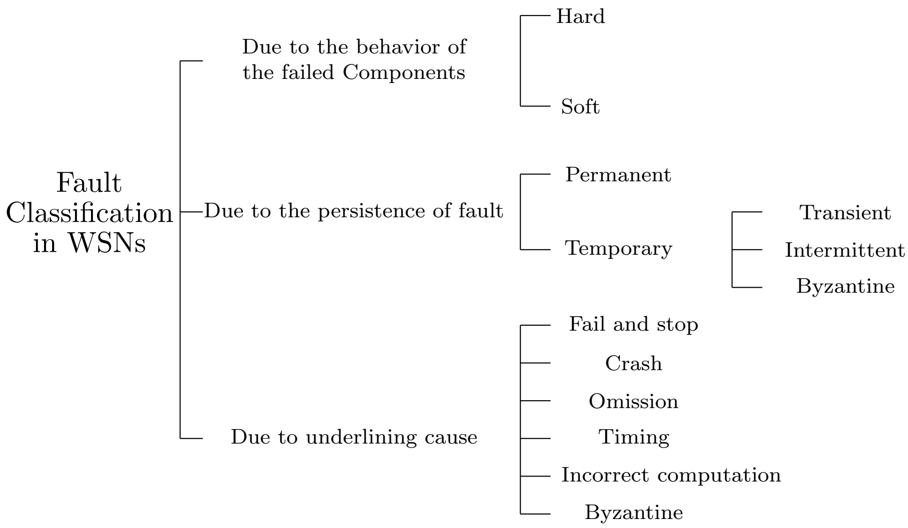


Figure 1.3: Fault Classification in WSNs

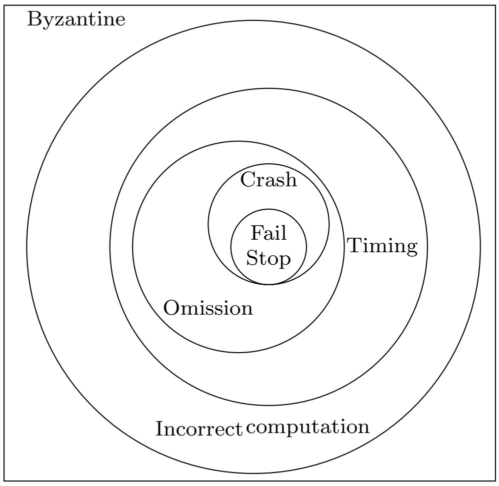


Figure 1.4: An ordered fault classification (adapted from [15]).

faults (the rate of change of the measured data does not match with expectations over an extended period of time), and drift faults (performance may drift away from the original calibration formulas). The falling battery voltage leads to calibration issues and cause the sensor to drift. Sensors with calibration error are treated as permanent faulty sensor node.

Sensor nodes may fail due to hardware problems such as poor connections or malfunctioning sensors or other embedded components. One of the prime causes of hardware faults is weather or environmental conditions. As reported by Szewczyk *et al.* [20], water contact with temperature and humidity sensors leads to a short circuit path between the power terminals which in turn causes high or low sensor readings. Electrical malfunctions may not be the only cause of hardware failure. For instance, the ion-selective electrode sensors used in soil deployments or sensors exposed to high radiation area are often prone to failures [21,22]. Such type of faults may appear continuously or intermittently.

Environmental noise is a common cause of sensor failure. Due to this, random errors are generated in sensor reading. Sensor reading is subjected to several sources of noise such as noise from external sources (electromagnetic interference, atmospheric perturbation), and hardware noise (low batteries) [23].

Residual energy left in the battery relative to the minimum operating power required for sensor operation is a crucial measure of sensor status [20, 21]. Low battery levels are not only an indication of remaining lifetime of a sensor node, but also it can also influence sensor readings from different perspectives and cause unreliable or faulty data. Ramanathan *et al.* [24] have experimentally shown that old battery can result in significantly erroneous data.

The path between source and destination in WSNs contains multiple wireless links (hops). The wireless links between sensor nodes are susceptible to wireless channel fading, which causes link failure. In addition, links may fail permanently or temporarily when the link is blocked by an external object, environmental changes, *etc.* Faults due to channel fading are transient and intermittent in nature. In this work, we have focused on the diagnosis of sensor nodes assuming that the diagnosis of link faults are taken care by underlying communication protocols.

Impact of the Sensor Fault over WSNs

Due to the presence of hard fault, the network may be partitioned into a number of sub networks, which results a break in the routing path [9]. In the presence of omission, timing, incorrect computation, spike and Byzantine fault, the existing network may not be partitioned into a number of sub networks. These faults yield degradation in the network performance. The presence of omission and timing fault imposes timing constraints on computations and produces correct values with an excessive delay. For example, an overloaded sensor node (e.g., cluster head) suffers with timing fault produces correct results with an excessive delay due to which other cluster head or sink declares it as a faulty sensor node and its information are ignored. This leads to reporting of erroneous information and degradation of network performance.

1.1.3 Fault Management in WSNs

The techniques used for handling the faulty sensor nodes are broadly categorized into the following types.

- *Fault prevention* : This technique is used to avoid the faulty sensor nodes reading so that overall performance of the sensor network remains as it is.
- *Fault identification / detection algorithms* : This technique is used to identify the presence of faulty sensor nodes in WSNs.
- *Fault diagnosis algorithms* : This technique is used to find the list of faulty and fault free sensor nodes in WSNs.
- *Fault tolerance mechanisms* : This technique is used to allow WSNs to continue its work or operation despite the occurrence of fault in WSNs.
- *Fault recovery mechanisms* : This technique is used to repair or recover the faulty sensor nodes during the network operation or at some later time in WSNs.
- *Fault isolation mechanisms* : In this method, the list of fault free and faulty sensor nodes are identified. Then, the list of faulty sensor nodes is separated from the network with an aim not to allow for participation in network operation.

The above fault management techniques are important to provide fault free information and continue for normal operation of WSNs. In this work, fault diagnosis algorithms has been mainly focused.

1.1.4 Performance Metrics

The performance of the fault diagnosis algorithms is measured in terms of the following parameters [7, 25].

1. **Diagnosis accuracy** is defined as the ratio between the number of faulty sensor nodes diagnosed as faulty and the total number of faulty sensor nodes present in the network.

2. **False alarm rate** is defined as the ratio of the number of fault free sensor nodes diagnosed as faulty to the total number of fault free sensor nodes present in the network.

3. **False positive rate** is defined as the ratio between the number of faulty sensor nodes diagnosed as fault free and the total number of faulty sensor nodes present in the network.

4. **Diagnosis latency** is defined as the maximum time required by the sensor nodes to diagnose the faulty node present in the network.

5. **Message exchange** is defined as the total number of messages exchanged over the network for fault diagnosis.

6. **Energy consumption** is defined as the total energy consumed by the network to identify the faulty sensor nodes present in the network.

7. **Network life time** is defined as the total number of data gathering rounds which will cause the first sensor node of the network to die due to energy depletion.

1.2 Motivation

Large-scale deployment of low-cost sensor nodes in inaccessible or hostile environments is an inherent property of WSNs. It is common for the sensor nodes to become faulty and unreliable due to natural calamities and environmental noise. The normal operation of a WSN suffers from faulty data since it decreases the judgment accuracy of the base station and increases the traffic and wastes the energy of sensor nodes [26]. Fault diagnosis appears to be a viable solution to these problems and serves as a tool that enhances data reliability, event reporting, effective bandwidth and energy utilization.

In most of the conventional fault diagnosis techniques devised for wired interconnected networks [27–34], and wireless networks [15,35,36] are not suitable for WSNs due to the following constraints.

- *Resource constraints.* Limited nodes processor power, communication bandwidth, small memory, and limited energy source are the constraints in WSNs. Since the message exchange is the only means of fault diagnosis and the energy consumed is proportional to the amount of traffic generated in diagnosing WSNs, a challenge for fault diagnosis in WSNs is how to minimize the energy overhead while keeping high diagnosis accuracy and low false alarm rate.

- *Random deployment.* Sensor nodes are randomly deployed by a human or a robot [2]. Fault-free sensor nodes may be wrongly diagnosed as faulty in a threshold-based diagnosis scheme [25, 37] if such schemes are applied to a sparse network or a randomly deployed WSNs having sparse areas.

- *Dynamic network topology.* In this scenario, sensor node densities show large spatio-temporal variations. Dissemination of diagnostic information in such dynamic networks is extremely challenging because network connectivity is a big issue. The ability of diagnosing faults decreases under this scenario, meaning that mobility significantly reduces the quality of the diagnosis returned by a diagnosis protocol [36].

- *Attenuation and signal loss.* The multi-hop communication in WSNs suffers from channel fading. In addition, applications like underwater, communications are established through transmission of acoustic waves [1]. In such applications, issues like limited bandwidth, long propagation delay, and signal fading make fault diagnosis more challenging.

The above said issues motivate the need to develop self fault diagnosis algorithms. Energy efficiency, low diagnosis latency, high diagnosis accuracy, and low false alarm rate are important goals in distributed fault diagnosis algorithm. If the self diagnosis algorithm is distributed, then it tries to minimize the amount of communication required by processing the data locally as much as possible. Therefore the proposed self diagnosis algorithms are distributed in nature where each sensor node accumulated the data from the neighbors and diagnose itself. The companion based statistical approach for fault diagnosis enhances the computation and communication overhead [4]. Since the mean and variance are not robust statistical measure, modified three sigma based robust diagnosis methods are proposed to improve the performance. Similarly, the intermittent fault diagnosis is more complicated as an intermittent faulty sensor node behaves faulty for a duration and behaves fault free in another duration of network operation. This motivates to model the intermittent fault behavior as Bernoulli distribution. The fault status is repeatedly tested by using robust statistical test and then predicts the intermittent fault status.

1.3 The Objective of the Research

In this thesis, new self fault diagnosis mechanisms have been proposed based on statistical approach to reduce the diagnosis overhead by maintaining high diagnosis accuracy, low false alarm rate, low false positive rate, low diagnosis latency and low communication overhead and low energy overhead which enhance the network performance. In particular, the objectives are as follows:

1. To design and evaluate distributed self fault diagnosis algorithm using neighbor coordination approach. An optimal threshold is to be devised using the normal Gaussian distribution function for effective self fault diagnosis in both sparse and dense WSNs.

2. To design and evaluate distributed self fault diagnosis algorithm using Newmann Pearson (NP) hypothesis testing method. An optimal threshold is to be derived with respect to network parameters.

3. To design and evaluate a robust self fault diagnosis algorithm using three sigma edit test and modified three sigma edit test which can diagnose the dense WSNs. The confidence interval of diagnosis accuracy, and false alarm rate is to be analyzed.

4. To design and evaluate robust distributed self intermittent fault diagnosis algorithm in WSNs using modified three sigma edit test where intermittently faulty behavior of the sensor nodes are studied using Bernoulli distribution.

5. The sensor's data model is proposed for fault diagnosis.

6. To validate the proposed distributed self fault diagnosis and existing algorithms in discrete event network simulator NS3 [38].

7. The efficacy of the proposed algorithms to be demonstrated by evaluating the performance parameters defined in Section 1.1.4.

1.4 Major Contribution

Chapter 1

Introduction to WSNs, overview of fault classification, fault management in WSNs are presented in this chapter. The motivation behind the energy efficient distributed self fault diagnosis algorithm over distributed fault detection and diagnosis method is outlined. The motivation of present research structure and the chapter wise presentation of the dissertation are also dealt in this chapter.

Chapter 2

This chapter provides a comprehensive overview of the related work done by different authors in the area of fault detection and diagnosis in WSNs. The main focuses are given to distributed fault detection and self fault diagnosis in WSNs.

Chapter 3

In this chapter, a novel distributed self fault diagnosis algorithm based on neighbor coordination (DSFDNC) approach is proposed by using the concept of the comparison model [10,39]. The performance analysis of the proposed DSFDNC algorithm has been carried out and has been shown that the new algorithm outperforms over the existing distributed fault diagnosis (DFD) [6] and improved distributed fault diagnosis (IDFD) [40] algorithms. Theoretical bound for the threshold used in DSFDNC is derived using statistical mechanisms.

Chapter 4

In this chapter, a distributed self fault diagnosis algorithm (DSFDHT) is proposed by using the concept of statistical hypothesis testing mechanism. The performance analysis of the DSFDHT algorithm has been carried out and shown that the algorithm outperforms over the DSFDNC, and existing DFD and IDFD algorithms. Theoretical bound for the threshold used in DSFDHT is derived using Neyman-Pearson hypothesis testing mechanism. An analysis of communication cost, total number of message exchanges and diagnosis latency are presented.

Chapter 5

Robust distributed self fault diagnosis algorithms (DSFD3SET) for WSNs based on modified three sigma edit test is presented in Chapter 5. The importance of robust three sigma edit test over other statistical methods like mean, median, and three edit test is presented here with an example. The robust performance of the algorithm is verified. This technique needs less communication overhead compared to DSFDHT, DFD, and IDFD algorithms and hence enhances the EC, NLT, and DL performance.

Chapter 6

In this chapter, a distributed self fault diagnosis algorithm is discussed to diagnose the intermittent faulty sensor nodes present in large scale WSNs. The performance of the proposed DSIFD3SET algorithm is measured after implementing in NS3. For diagnosing the intermittently faulty sensor nodes, modified three sigma edit test is used, and the intermittent faulty behavior of the sensor nodes is studied by using Bernoulli distribution.

Chapter 7

Finally, Chapter 7 outlines the conclusion of the work. It also discusses the achievements and limitations of the results obtained. This chapter ends with future scopes for this work.

1.5 Thesis organization

In this dissertation, four self fault diagnosis algorithms, namely DSFDNC, DSFDHT, DSFD3SET, and DSIFD3SET are proposed to diagnose the hard and soft faulty sensor nodes in WSNs.

- The DSFDNC algorithm is based on a realistic fault and data model. The accuracy and completeness of the DSFDNC algorithm are evaluated by modeling the error, assuming to follow normal Gaussian distribution. The simulation result shows that the performance of the proposed algorithm is improved compared to that of DFD and IDFD algorithms.

- Event detection using NP hypothesis testing is an important problem in statistics. A similar idea is incorporated to diagnose a faulty sensor node present in WSNs based on which the DSFDHT algorithm has been proposed. The algorithm is developed based on similar data model used in the DSFDNC. The performance of the algorithm is improved when the average degree of sensor nodes in the network is less.

- A modified three sigma edit test based distributed self fault diagnosis algorithm for large scale WSNs is proposed to make the algorithm robust. The DSFD3SET algorithm diagnoses the faulty nodes with less number of message exchanges. The performance is better than the DSFDNC and DSFDHT algorithms in dense WSNs.

- To diagnose the intermittent faulty sensor nodes in WSNs the DSIFD3SET algorithm is proposed. The intermittent faulty behavior is modeled by using the Bernoulli distribution function. The fault status of sensor nodes is diagnosed repeatedly by using modified three sigma edit test. The performance of the DSIFD3SET algorithm is compared with the distributed intermittent fault diagnosis (DIFD) [25] algorithm.

1.6 Conclusion

This chapter provides a brief introduction to WSNs, cause of fault occurrence, an overview of fault types. It also systematically outlines the scope, the motivation, and the objectives of the thesis. A concise presentation of research work carried out in each chapter and the contribution made in the thesis have also been presented. In essence, this chapter provides a complete overview of the total thesis in a condensed manner.

Chapter 2

Background and Literature Survey on Fault Diagnosis Algorithms in Wireless Sensor Networks

Chapter 2

Background and Literature Survey

In this chapter, an exhaustive literature survey on fault diagnosis is presented. The fault diagnosis approaches are classified into various groups based on different criteria such as fault diagnosis procedure, number of sensor nodes participating in the diagnosis process, implementation and fault type.

2.1 Introduction

The fault diagnosis in networks is an important area of research since 1967 known as system level diagnosis [41]. The fault diagnosis algorithms have been mainly developed for multi processor and multi computers system which depend on various system model and fault model. The system model presents the characteristic of the network and communication among the system components and the fault model specifies the behavior of the system components when one or more fault types occur in the system. According to the system level diagnosis, a system can be decomposed into a number of units and each unit is capable of testing other units. The fault diagnosis algorithms available for wired network are not suitable for wireless sensor network due to the characteristics of sensor nodes and wireless communication medium.

The rest of the chapter is organized as follows. The taxonomy of fault diagnosis and diagnosis in wireless sensor networks is presented in Section 2.2. The system and fault model for diagnosis algorithms are summarized in Section 2.3. Section 2.4 presents the methods based on which the faulty and fault free sensor nodes are diagnosed. Finally, this chapter is summarized in Section 2.5.

2.2 Taxonomy of Fault Diagnosis in WSNs

The fault diagnosis algorithms in WSNs are broadly categorized into different types such as test type, comparison based, neighbor coordination, probabilistic, statistical, rule based, automaton, number of sensor nodes involvement in the diagnosis process, observation time, fault types, and soft computing approaches which are summarized in Figure 2.1. Based on the sensor node involvement, the fault diagnosis algorithms are further classified into three types such as centralized, distributed and self diagnosis.

In a centralized approach, a single sensor node keeps track of the entire network fault status. It is an ultra reliable sensor node which remains fault free for the entire duration of network operation [42]. This approach needs more communication overhead. However, the key advantage of this approach is that the information required are available in a central place where all other sensor nodes can access and synchronize with this sensor node. Similarly, in distributed diagnosis, two or more numbers of sensor nodes are involved in fault diagnosis [43]. When all the sensor nodes participate in the diagnosis process to identify their fault status, this is known as self fault diagnosis [44, 45]. Both the centralized and distributed approaches use more number of messages using multi hop communication. Whereas the self diagnosis approaches are widely used in sensor networks due to the fact that every sensor node exchange the messages in its neighborhood and take the advantage of the shared nature of wireless communication medium.

The distributed approaches are further divided into two subcategories based on the network connectivity such as partitionable and non partitionable network [9,35]. The methods used for detecting the faulty sensor nodes are classified as test based, neighbor co-ordination, statistical, probabilistic, and soft computing based approaches. The detail description about all these approaches is discussed in subsequent sections of this chapter.

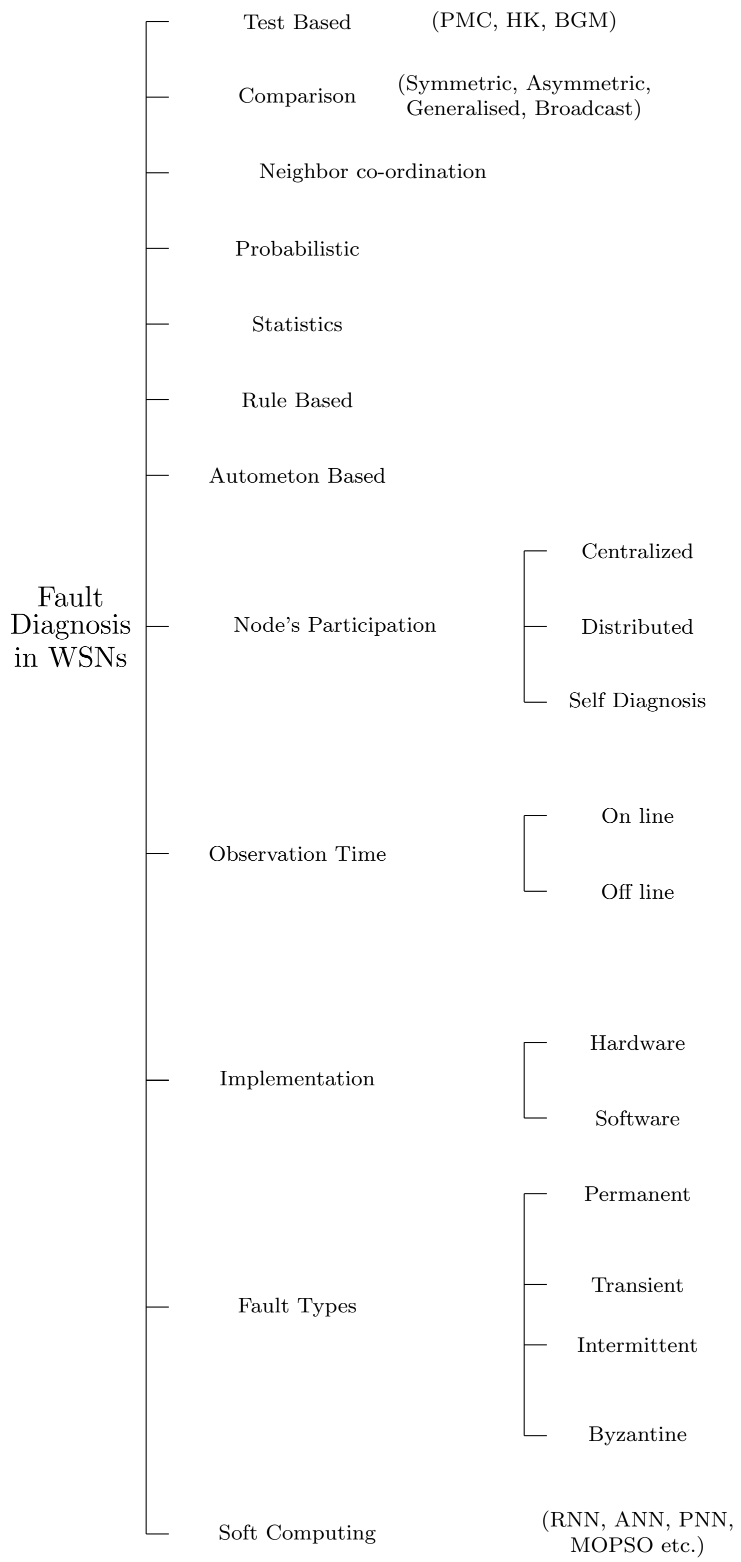


Figure 2.1: Classification of fault diagnosis in WSNs

2.3 System and Fault Model for Fault Diagnosis Algorithms

The existing fault diagnosis algorithms have been devised under the assumption of different types of system and fault models. The system model characterizes various features of a system such as network topology, communication system protocol, and interfaces. The fault model characterizes different types of faults and their behavior in the system [9,35,46]. The system model can be partitionable and non partitionable network, which are summarized in this section.

2.3.1 Fault Diagnosis in Non-Partitionable WSNs

In this approach, the algorithms assume the entire network as a single connected component. The number of hard faulty sensor nodes present around any sensor node s_i is $D-1$, where D represents the minimum degree of the sensor node s_i [46]. As the soft faulty sensor nodes allow the normal network operation, they do not affect the connectivity of the sensor network.

2.3.2 Fault Diagnosis in Partionable WSNs

Elias *et al.* [35] have proposed a fault diagnosis algorithm which handles both crash and timing faulty sensor nodes. WSNs are partitioned into an arbitrary number of sub networks. To identify the crash faulty sensor nodes, each sensor node tests their communication links to judge the faulty behavior. Each sensor node also keeps a local view of the network topology along with the time stamp of each communication link. This requires huge memory in each of the sensor nodes.

The algorithm is validated in terms of bounded diagnostic latency, bounded start up and accuracy. Each sensor node in WSNs plays as a tester and tested sensor node. The communication link among the tester and tested sensor node is known as tested link. A sensor node s_i tests its neighbors for a time interval and in the next time interval, neighboring nodes Neg_i test the sensor node s_i. This is based on the assumption that if the sensor node s_i tests all its neighbors successfully, then its neighbors are not suffering with crash or timing faults. This algorithm needs Q number of tests at a particular time instant, where Q is the number of communication links available to the sensor node. By this, extra communication and computing overhead is reduced.

Barooah *et al.* [9] addressed the crash fault which leads the WSNs into multiple numbers of connected components. Each component is obtained from a set of links known as a cut. The technique through which cuts are detected is known as cut detection algorithm. Cuts occurred when number of crash faulty sensor nodes of a particular sensor node s_i exceeds the degree of the sensor node s_i. This partionable network may yield due to following reasons [9].

- The routing path might experience a break
- Sensing area might experience a leak
- The batteries of some sensor nodes might be depleted
- Requiring more relay sensor nodes
- The sensor nodes wear out after a long period of time

The cut detection algorithm was initiated by any arbitrary sensor node present in the network, which is known as the source node. The algorithm has two phases. At the beginning phase it decides when a cut occurred with the sensor node s_i which will separate the sensor node from source node or not. In the later stage, it considers where the cut occurred with sensor node s_i.

In a partitionable WSN, the self fault diagnosis algorithms are suitable because each sensor node diagnose itself with the help of neighboring sensor node's data. If the network is partitioned, each sensor node independently diagnose its own status, though with a degradable performance, because the degree of the network changes after the partition.

2.4 Fault Diagnosis Algorithms

2.4.1 Test Based

In this approach, each sensor node s_i acts as tester (tests other nodes) as well as a testee node (tested by other nodes) [41, 46, 47]. Each sensor node s_i assigns a test task or test sequence t_i to all its neighbors Neg_i. Upon receiving the test task or test sequence t_i, each of the neighboring node s_j ($s_j \in Neg_i$) evaluate the test task t_i and returns the response message or response sequence to sensor node s_i. The testing node s_i outputs a test outcome $c_{ij} = 1$, if the actual response message or sequence mismatches the expected one; otherwise $c_{i,j} = 0$ and informs the test outcome c_{ij} to the central processor for which it needs multi hop communication.

The collection of all test outcomes between every pair of sensor nodes is known as a syndrome [46–48]. Based on the syndrome the sensor node's fault status is determined. For generating the syndrome, each sensor node s_i needs minimum two message exchanges (test and response messages) over the network which needs maximum $2N$ message exchanges. For generating the final fault status, ND messages are exchanged over the network where D is the diameter of the N nodes network. This approach does not depend on spatial and temporal relationship among the nodes. This approach is applicable to multiprocessor systems. As energy is a constraint in WSNs [39], the test based approach is not suitable.

Preparata *et al.* (PMC) model [41] is the first model which is a one-step f-diagnosable system that can identify maximum f faulty sensor nodes from a given syndrome in one step. All sensor nodes are participating in fault diagnosis process to test each other. It is assumed that the test outcomes are correct if the testing unit is fault free; otherwise, the outcomes are unreliable. Directed graph is generated based on the number of nodes participated in diagnosis process and each one is connected with others with a directed edge which is labeled by test results. Those test results are generated by the group of tester and testing nodes. In this fault diagnosis model, each node starts its diagnosis procedure by sending the test syndrome to the base station. According to the collection of all test outcomes, the fault status of every sensor node can be identified at the base station.

Kreutzer and Hakimi (HK) model in [49] and Meyer *et al.* (BGM) model in [50] are the variation of PMC model. In those models, each sensor node is assigned with a set of sensor nodes which may or may not be the neighbors of sensor node s_i. Then, the sensor node s_i assign test task t_i to those sensor nodes and waits to receive a test response from them. After collecting the test response from all, central node diagnoses the network by analyzing the test outcomes.

2.4.2 Comparison Based

In comparison model, same task is assigned to multiple numbers of neighboring nodes Neg_i (other than the sensor node which are already tested). Each of the neighbor computes their respective task and send back their result. Then, the sensor node s_i computes the status of the neighboring nodes Neg_i by analyzing the result. This model was first proposed by Malek in MM [46] and Hakim *et al.* in MM* [47]. In MM model, a sensor node s_i tests any pair of sensor nodes s_j, and s_k (may or may not be neighbors) present in the network by sending same test task t_i to them. The source node s_i analyze the result r_i received from them. If both the task results are equal, then, the sensor node s_i concludes that nodes s_j, and s_k are fault free otherwise faulty.

But, in MM* model, a sensor node s_i can test any pair of sensor nodes s_j, and s_k including the sensor node which are already tested. The nodes compute the task and send the result back to the testing sensor node s_i. The sensor node s_i computes the status of the sensor node s_j, and s_k by analyzing their results r_j, and r_k respectively. This model is applied over only the hard or soft faulty sensor nodes present in the network. The MM and MM* models differ from each other based on a test involving the pair of faulty sensor nodes. In the symmetric model (MM*), both test outcomes (0 & 1) are possible where as in the asymmetric model (MM) two faulty nodes always give mismatching outputs i.e. 1.

Unlike MM and MM* model, Sengupta and Dahbura presented a generalization of invalidation and comparison models by introducing a new model, known as the generalized comparison model (GC), in which the comparator sensor node can be one of the two sensor nodes under comparison [5]. Blough and Brown introduced a distributed diagnosis algorithm using a generalized comparison model [51]. They developed the first broadcast comparison model (BC), in which two nodes under comparison broadcast their outputs to all sensor nodes in the system.

Identifying all faulty sensor nodes present within the sensor network using the comparison model is an NP hard problem [52]. When the problem is reduced to t-diagnosable problem where t is the maximum t faulty sensor nodes can be diagnosed, then the problem is termed as a polynomial time algorithm. Albini *et al.* [53] introduced the generalized distributed comparison-based (GDC) model which is based on the asymmetric comparison model which requires that a fault-free sensor node execute tasks within a bounded time duration. The comparison model is supposed to be the most practical model for various diagnosis systems such as WSNs, MANET, and other wireless networks.

Table 2.1: The comparison of different fault models

	MM	**MM***	**GC**	**BC**	**GDC**
Model type	Asymetric	Symmetric	Both	Both	Asymetric
Comparator node	Non participating node	Any	Any	Any	Any
NP Hard	$\surd$	$\surd$	$\surd$	$\times$	$\times$
communication type	one to one	one to one	one to one	Broad cast	one to one
Time duration	Unbounded	Unbounded	Unbounded	Unbounded	Bounded

2.4.3 Neighbor Coordination Based

In neighbor coordination based distributed fault diagnosis algorithm, each sensor node s_i compares its own sensed data x_i with its neighbors data and send the results (in terms of 0 or 1) back to its neighbors [6, 40, 54]. The probability of sensor node s_i's fault status is computed based on majority voting performed with its neighbors data. Each likely fault free sensor node is identified as fault free sensors by a rigid criteria as described in Equation (2.1) , where FS_i is the fault status of the sensor node s_i. This approach is based on spatial relationships between sensor nodes.

$$FS_i = \begin{cases} \text{GD} & \text{if } \sum\limits_{j \in Neg_i} R_{ij} < 0.5D \\ \text{FT} & \text{otherwise} \end{cases} \tag{2.1}$$

The faulty sensor node's data are spatially uncorrelated while the fault free sensor data are spatially correlated. Sensor reading x_i is similar to sensor reading x_j when

$|x_i - x_j| < \delta$ and δ is expected to be a small number (as nearer sensors have similar reading). Hence the fundamental principle of this approach is to compare a sensor node s_i's data with $s_j \in Neg_i$ and find $R_{ij} \in \{0, 1\}$. As shown in Figure 2.2, $R_{ij} = 0$ if $|x_i - x_j| < \delta$. Otherwise, $R_{ij} = 1$. This approach estimates the faulty state of the sensor node s_i by comparing the number of 0s with a predefined threshold or by using Equation (2.1). This approach is illustrated in Figure 2.2.

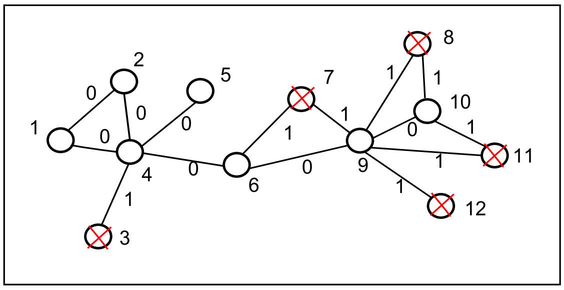


Figure 2.2: Illustration of comparison result. Crossed sensor nodes are faulty (adapted from [25]).

Chen *et al.* [6] proposed a distributed fault diagnosis algorithm (DFD) to identify the soft faulty sensor nodes present in WSNs. It uses local comparisons with a modified majority voting scheme to identify the faulty sensor nodes. In DFD, each sensor node s_i makes a decision based on comparisons between its own sensed data x_i with its one-hop neighbors data.

The algorithm consists of four test phases. In the first phase, a test result $R_{ij} \in \{0, 1\}$ is generated based on its neighbor data using two variables, namely $m_{ij}^{T_l}$ and $\Delta m_{ij}^{\Delta T_l}$, and two predefined threshold values Φ_1 and Φ_2. The measured difference between the sensor node s_i and s_j from time T_l to T_{l+1} is defined as

$$\Delta m_{ij}^{\Delta T_l} = m_{ij}^{T_{l+1}} - m_{ij}^{T_l} = (x_i^{T_{l+1}} - x_j^{T_{l+1}}) - (x_i^{T_l} - x_j^{T_l})$$

where $x_i^{T_l}$ is the reading of the sensor node s_i at time T_l.

For any sensor node $s_j \in Neg_i$, the sensor node s_i first set R_{ij} to 0. Then next calculates $m_{ij}^{T_l}$. If $|m_{ij}^{T_l}| > \Phi_1$ then it calculates $\Delta m_{ij}^{\Delta T_l}$. The comparison test result R_{ij} is set to 1 if $|\Delta m_{ij}^{\Delta T_l}| > \Phi_2$. If R_{ij} is 0, most likely either both sensor node s_i and s_j are good or both are faulty. Otherwise, if R_{ij} is 1, the sensor node s_i and s_j are most likely in different status. In this approach, for any sensor node s_i, its test results with each sensor node in the neighbor set Neg_i is obtained. If there are more than $\lceil |Neg_i|/2 \rceil$ sensor nodes whose comparison test results are 1 in Neg_i, then initial diagnosis status (i.e., tendency value $Tend_i$) of sensor node s_i is possibly faulty (*LT*), otherwise, it may be possible normal (LG), i.e.,

$$Tend_i = \begin{cases} \text{LT} & \text{if} \sum\limits_{s_j \in Neg_i} R_{ij} \geq \lceil |Neg_i|/2 \rceil \\ \text{LG} & \text{otherwise} \end{cases} \tag{2.2}$$

where $|Neg_i|$ represents the number of one-hop neighbors of the sensor node s_i. Each sensor node s_i sends its tendency value to all its neighbors. When the initial diagnosis status of all sensor nodes in the WSN is obtained, in the second round of test of DFD algorithm, the number of LG nodes whose test result with the sensor node s_i is 1 is subtracted from the number of LG nodes whose test result with the sensor node s_i is 0. If the result is greater than or equals to $\lceil |Neg_i|/2 \rceil$, then the sensor node s_i is detected as fault-free. That is $\forall s_j \in Neg_i$ and $Tend_j = LG$, $\sum(1-R_{ij}) - \sum R_{ij} = \sum(1-2R_{ij})$ must be greater or equal to $\lceil |Neg_i)|/2 \rceil$ to detect the sensor node s_i as fault-free. This can be defined as

$$s_i = \begin{cases} \text{fault-free(GD)} & \text{if} \sum\limits_{s_j \in Neg_i, Tend_j = LG} (1-2R_{ij}) \geq \lceil |Neg_i|/2 \rceil \\ \text{Undetermined} & \text{otherwise} \end{cases} \tag{2.3}$$

A sensor node s_i that has failed to pass the threshold test of Equation (2.3) is marked as *undetermined* and follows a third round of test. All the *undetermined* nodes repeatedly check for $\log n$ times in the best case ($\sqrt{N}$ in the average case and N times in the worst case) if one of its neighbors is determined to be fault-free. If such a sensor node exists and $R_{ij} = 0(1)$ then the sensor node s_i is detected as fault-free (faulty).

If still ambiguity occurs, in the fourth round of test, the sensor's own tendency value determines its status. For instance, if the status of the sensor node $s_j, s_k \in Neg_i$ is determined as fault free (i.e., $Tend_j = Tend_k = GD$), the sensor node s_i is marked as *undetermined* and $R_{ji} \neq R_{ki}$ then the sensor node s_i will be detected as fault-free (faulty) if $Tend_i = LG(FT)$. The time and message complexity of this approach is $O(D)$ and $O(N)$ respectively.

Jiang [40] claims an improvement over the DFD algorithm by introducing an improved distributed fault diagnosis algorithm (IDFD Algorithm). In this approach, a sensor node s_i first set R_{ij} to 0 for any sensor node $s_j \in Neg_i$. IDFD algorithm then calculates $m_{ij}^{T_l}$ and if $|m_{ij}^{T_l}| > \Phi_1$ then comparison test result R_{ij} is set to 1. If $|m_{ij}^{T_l}| \leq \Phi_1$ then it calculates $\Delta m_{ij}^{\Delta T_l}$. The comparison test results R_{ij} is set to 1 if $|\Delta m_{ij}^{\Delta T_l}| > \Phi_2$. This algorithm then follows Equation (2.2) to determine the initial diagnostic status (i.e., LG or LT) of the nodes. In this approach, for any sensor node s_i and the sensor nodes in Neg_i whose initial diagnosis status is LG, if the sensor node whose test result with the sensor node s_i is 0 is not less than the nodes whose test result is 1, then the status of the sensor node s_i is GD. Otherwise, the status of the sensor node s_i is FT. Alternatively, this can be explained as

$$s_i = \begin{cases} \text{GD} & \text{if} \sum\limits_{s_j \in Neg_i), Tend_j = LG} R_{ij} < \lceil |Neg_{iTend_j = LG}|/2 \rceil \\ \text{FT} & \text{otherwise} \end{cases} \tag{2.4}$$

If there are no neighbor nodes of the sensor node s_i whose initial diagnosis status is LG, and if the initial diagnosis status $Tend_i$ of the sensor node s_i is LG, then this algorithm sets the status of the sensor node s_i to GD, otherwise to FT. There will be four types of message exchanges to achieve the fault diagnosis. As time and message complexity is high, this approach is not suitable for energy constrained WSN. The time and message complexity of this approach are $O(D)$ and $O(N)$ respectively, where N is the total number of sensor nodes and D is the maximum degree of the network.

Hsin *et al.* [54] suggested a two-phase time out mechanism based neighbor coordination approach in which two timer values C_1 and C_2 are used. In the first phase, a sensor node waits for its neighbors to update information regarding the faulty nodes. In the second phase, it consults with its neighbors to reach a more accurate decision. To increase the efficiency of the algorithm, two timers are maintained for monitoring a sensor node. If a sensor node $s_j \in Neg_i$ does not receive any packets from s_i before $C_1(s_i)$ expires, s_j activates the second timer $C_2(s_i)$. During the second time period, s_j will query the common neighbors regarding the status of the sensor node s_i and take a decision accordingly. The time and message complexity of this approach is $O(D)$ and $O(N)$ respectively.

The majority-voting techniques have the potential to enhance the diagnostic performance in terms of diagnosis accuracy, false positive rate and false alarm rate. The performances of these techniques are very low for the sparsely deployed network. However, this approach gives better performance over the densely deployed network, i.e. high average degree network.

Table 2.2: The comparison of different neighbor coordination approaches

Algorithms	**Time complexity**	**Message complexity**	**Spatial Relationship**	**Temporal Relationship**	**Majority voting**
DFD Algo [6]	$5D$	$5N$	$\surd$	$\times$	$\surd$
IDFD [40]	$3D$	$3N$	$\surd$	$\times$	$\surd$
Hsin *et al.* [54]	$2D$	$2N$	$\times$	$\surd$	$\times$

There can be a serious disadvantage of voting based fault diagnosis scheme, where sensor readings from different neighbors are not reliable. For example, an intruder in the deployment environment may misguide a sensor node to send faulty readings to its neighbors, which are usually the assumption of these approaches. There are major shortcomings with this approach given in Table 2.2, such as (i) computational and communication overhead is high, (ii) majority voting is not suitable for unreliable sensor networks. In order to overcome these difficulties, the self fault diagnosis approaches are presented in this dissertation.

2.4.4 Statistical Approach

The statistical approaches such as mean, median, weighted mean, and weighted median use the spatial correlation of sensor measurements to identify the faulty sensor nodes in WSNs [55–57]. A sensor node can detect itself as faulty or fault free by using statistical tests. All the statistical methods are published based on mean, weighted mean, median, and weighted median. These methods have less diagnosis accuracy, and false alarm rate.

Gao *et al.* [55] have proposed a self fault diagnosis algorithm based on the weighted median concept which does not require any majority voting scheme. Each sensor node s_i collects sensed data x_i from their neighbors Neg_i. So, it does not require any extra message transmission or reception for fault diagnosis. By this it can save some amount of energy of the sensor nodes. The authors define a decision function $f(x_i, \hat{x}_i)$ to detect soft faulty sensor nodes in the network which is defined as follows.

$$f(x_i, \hat{x}_i) = \begin{cases} 1 & \text{if } |\frac{x_i - \hat{x}_i}{x_i}| > \xi \\ 0 & \text{otherwise} \end{cases}$$

where $\hat{x}_i$ is the weighted median of its M one-hop neighbor's data. Here the measured values are x_j, and their corresponding weights are $\lambda_j (j = 1, \cdots, M)$. These weights represent their corresponding confidence degrees. Assuming that the sensed data x_j is in increasing order, the weighted median is formulated as

$$\hat{x}_i = MED\left\{\lambda_j \diamond x_j|_{j=1}^{M}\right\}$$

where MED is the median operation, which outputs the middle of a distribution. The operator $\diamond$ characterizes the duplication operation such that the sensed data x_j is duplicated for λ_j times.

Nodes are able to receive, send, and process when they are faulty, i.e., soft faulty. Total number of message exchanges to detect all the faulty nodes need $O(N)$, where N represents the total number of sensor nodes deployed in the network. As median operation is applied over a sorted data the time complexity of this approach is $O(D \times \log D)$ where D represents the maximum degree of the network. The fault diagnosis algorithm is included within the normal workload of the sensor network. Thus, it does not require any extra energy for fault diagnosis.

A localized fault diagnosis algorithm in WSNs is analyzed by Ding *et al.* [56]. It is a distributed fault diagnosis algorithm, where each sensor node s_i compares its own sensed data x_i with the median of neighbor's data in order to determine its own status. The performance of localized diagnosis is limited due to the non-uniform nature of the node in WSN. The time complexity of this approach is same as [55].

Sai *et al.* [57] have proposed a fault diagnosis algorithm based on the weighted mean operation which does not require the majority voting scheme. A decision function $f(x_i, \hat{x}_i)$ is defined to detect soft faulty sensor nodes as follows.

$$f(x_i, \hat{x}_i) = \begin{cases} 1 & \text{if } |x_i - \hat{x}_i| > \theta \\ 0 & \text{otherwise} \end{cases}$$

where the $\hat{x}_i$ is defined as given in (2.5).

$$\bar{x}_i = AVG(x_j)|_{j=1}^{M} = \frac{\sum_{j=1}^{M}(\lambda_j x_j)}{\sum_{i=1}^{M} \lambda_j} \tag{2.5}$$

$\bar{x}_i$ is the weighted mean of its M one-hop neighbors' data. $\lambda_j (j = 1, \cdots, M)$ is their corresponding weights. These weights represent their corresponding confidence degrees, which are random number between λ_{min} and λ_{max}.

The statistical based approaches are suitable for terrestrial application domain where sensor nodes are densely deployed and there exist a spatial correlation between the sensor readings. A comparison study of different statistical approaches are summarized in Table 2.3.

Table 2.3: The comparison of different Statistical approaches

Algorithms	**Time complexity**	**Message complexity**	**Spatial Relationship**	**Temporal Relationship**	**Majority voting**	**Approach**
Gao *et al.* [55]	$D \log D$	N	$\times$	$\times$	$\times$	Weighted Median
Ding *et al.* [56]	$2D$	$2N$	$\times$	$\times$	$\surd$	Median
Sai *et al.* [57]	D	N	$\times$	$\times$	$\times$	Weighted Mean

2.4.5 Probabilistic Approach

Nandi *et al.* [43] have proposed a model based fault diagnosis approach in which they consider the soft faulty sensor nodes only. For identifying those faulty sensor nodes Bayesian model is used. In this approach the entire region of interest was partitioned into a number of hexagonal shaped sub regions where sensor node was placed at the center of the hexagonal region. Here each sensor node s_i is capable to send their data directly to the base station where the faulty sensor nodes are declared based on Bayesian rule. As the central node is also subjected to fault, this method has overhead between the central sensor node as well as a boundary sensor node to diagnose the fault. This is a centralized approach and it needs $O(N)$ message exchanges over the network and $O(N)$ time complexity to achieve the diagnosis.

Nandi *et al.* [58] have proposed a classical hypothesis testing based topology dependent fault diagnosis algorithm. Entire regions of interest (ROI) was partitioned into a number of sub squares where the sensor node s_i is placed at the center of the ROI. The author assumes that all the sensor nodes are used to detect an event occurred around its surrounding regions. Faulty nodes are detected based on the error probability which is calculated by the base station based on the Neyman-Pearson Most Powerful (MP) test. As each sensor node s_i sends its sensed data x_i directly to the base station, it needs $O(N)$ message exchanges over the network. This approach puts fewer bottlenecks over the channel as this approach does not require multi-hop communication. However, it depletes the battery so rapidly due to the direct communication between base station and sensor node as energy is proportional to distance.

Lau *et al.* [59] have proposed a probabilistic based fault diagnosis algorithm in WSNs which uses extra resource consumption. End to end transmission time and Bayes classifier was used to diagnose the hard and soft faulty sensor nodes in the network. List of faulty nodes is diagnosed by the base station by gathering the regular information from the sensor nodes. This approach does not consume energy of sensor node for fault diagnosis. Because, the fault diagnosis mechanism is based on, an end to end transmission time of individual packet coming to the base station from sensor nodes. The fault diagnosis accuracy is more as it uses the Bayes classifier. It transmits maximum N packets over the network due to which its time complexity was $O(N)$. As normal data are used for diagnosis purpose there is no extra overhead for diagnosis.

Though the probabilistic methods are suitable for diagnosis in terms of computation and communication overhead, the base station leads to a centralized bottleneck which may not be reliable. In fact, if the central node is faulty, the entire diagnostic process results in catastrophic situations. In order to avoid the centralized bottleneck, the distributed self fault diagnosis approach is proposed.

2.4.6 Rule Based

Rule or learning based fault diagnosis method is used for diagnosing the soft faulty sensor nodes [18]. This technique is used to identify only constant, noise and short soft faulty nodes. Linear least square estimation (LLSE) method is used under the estimation technique. Forecasting based approach and hidden Markov model uses the time series analysis to predict the fault status of sensor nodes in a wireless sensor network. The accuracy of the rule based approach is more compared to other three approaches as mentioned above. To the best of our knowledge, the diagnosis of hard, transient, intermittent and byzantine faults in sensor networks using rule based approach have not been addressed in the literature though the rule based techniques are energy efficient.

2.4.7 Automaton Based

Liu *et al.* [44] have proposed a finite state machine based self fault diagnosis algorithm to diagnose the hard faulty sensor nodes present in the network. In their approach, the hard faulty sensor nodes occur only due to low battery power or system reboot, a neighboring node detects that a sensor node is dead or low link quality due to interference, and high retransmission ratio. The approach puts message overhead on the network. It also includes high transmission and computation cost.

Cellular automaton based fault diagnosis method has been discussed by Banerjee *et al.* [3]. In their approach, they establish a spatial and temporal correlation of sensing information based fault diagnosis approach which efficiently diagnoses the faulty sensor nodes.

2.4.8 Soft Computing Based

Soft computing based approaches are used for diagnosing the soft and hard faulty sensor nodes in the sensor network. The characteristics of the sensor nodes are used to diagnose the possible fault and fault free sensor nodes present in WSNs. The most prominent evolutionary or heuristic approaches like neural networks [52, 60], perceptron neural network [61], multi objective particle swarm optimization

[62], genetic algorithm [63], back propagation neural network [64], support vector machines [65], and radial basis function based neural network [66] etc. are used for fault diagnosis in WSNs.

Jabbari *et al.* [67] have proposed an Artificial Neural Network (ANN) based fault diagnosis algorithm in which faulty sensor nodes are diagnosed based on analysis of sensed data generated by individual sensor node. This approach follows two phases such as residual generation and verification. For residual generation, it uses the generalized regression neural network architecture and for residual verification, kernel-based learning method is applied.

Azzam *et al.* [68] have introduced a Recurrent Neural Network (RRN) based fault diagnosis in WSNs. As the RRN has the ability to capture and model the dynamic properties of nonlinear systems, this approach uses the model to represent a sensor node, the node's dynamics, and interconnecting with other nodes. This approach assumes that there is one sensor per sensor node where the sensor nodes are viewed as small dynamic systems with memory like features. The introduced ad-hoc RRN is analogous to WSN systems with confidence factors ($0 < CF_{ij} < 1$) between sensor nodes s_i and s_j. The overall modeling process is divided into two phases such as the learning and production phase. In the learning phase, the neural network adjusts its weights that correspond to the healthy and F faulty models. The production phase compares the current output of the sensor node with the output of the neural network. The difference between these two signals is the basis to detect a sensor's fault status. Barron *et al.* [69] implemented this approach on Moteiv's Tmote Sky platform with TinyOS operating system.

Elhadef *et al.* [52] have proposed a soft computing based system level fault diagnosis approach in multiprocessor and multicomputer systems. They model the fault diagnosis problem as two set classification problem in which one set is called hard faulty nodes set and another one is fault free nodes set. Here the entire fault diagnosis process is partitioned into two phases. In the first phase, the simple [46] and generalized [5] comparison models are used for generating the partial syndromes. These partial syndromes are generated by assigning the same task to any two pair of nodes and they evaluate the task independently. After evaluation, based on the agreement and disagreement on their results, a partial syndrome is generated. This task is done in a distributed manner. After this phase was over all the faulty nodes are diagnosed centrally at the base station. For diagnosing the faulty nodes, it follows the Back Propagation Neural Network (BPNN) approach.

These approaches such as ANN [67], PNN [68], and BPNN [52] based fault diagnosis put computational (processing time), communication (number of message exchanges), and resource (energy, memory) burden on WSNs. These approaches are not suitable for dynamic WSNs. The set of faulty nodes are determined through a designated sensor node which may not be a feasible solution for WSNs. These approaches need more memory as it operates on large data set.

2.4.9 Node Participation Based

The fault diagnosis processes are classified into three subcategories based on the number of nodes involvement during the diagnosis process. If a single node is responsible for fault diagnosis, then the fault diagnosis is termed as centralized diagnosis. In distributed diagnosis, two or more selective nodes are responsible for fault diagnosis. The self diagnosis is an approach in which each sensor node in WSNs is responsible for actively participating in the diagnosis process.

Centralized Diagnosis

In a centralized approach, a central node with high computation capability is responsible for diagnosing the fault status of every sensor node present in the network. For detecting the fault status it assigns one or more task to all the sensor nodes. These nodes receive the task, execute the task and send the response to the central node. Based on the response message the central node decides the status of each sensor node [42].

Guo *et al.* [70] deal with soft fault which is also called data faults in WSNs. FIND is a sequence-based fault diagnosis approach for identifying data faulty nodes in sensor networks. One kind of data can be received signal strength indication ($RSSI$) value of a sensor node. This approach assumes that the $RSSI$ value of receiving packets decreases as distance increases. The $RSSI$ value is different for different nodes. The algorithm is based on four phases of activity. In the first phase, entire rectangular terrine (sensor field) is partitioned into a number of sub regions by using perpendicular bisector. In the second phase, each of the sensor node sense the environment and send their data to the base station, which needs $O(N)$ data exchange over the network. After receiving the data, the base station analyzes the packets based on $RSSI$ value and estimate a distance sequence which is known as detected sequence. Then it uses any longest common subsequence approach to identify the matched sequence present in the data base which is known as estimated sequence. Finally, it analyzes the detected sequence with estimated sequence based on the ranking difference to identify the faulty node present in the network. The performance of the diagnosis algorithm is measured in terms of false positive and negative rate. The time complexity of the algorithm is $O(N^2)$.

Ramanathan *et al.* [71] have assumed that all functioning nodes present in the sensor network are responsible for generating a kind of traffic (i.e. routing updates information, time synchronization beacons, or data). This information is periodically transmitted to the sink node which monitors the traffic and establishes a statistical relationship between packets generated by sensor nodes. This statistical data is used for detecting a failure node present in the network and triggers the fault diagnosis method when a node generates less monitored traffic than expected. It uses multi hop communication and needs $O(N \times N_d)$ messages, where N is the total number of sensor nodes and N_d is the diameter of the senor network. The centralized approaches have following demerits.

1. For keeping status information of N sensor nodes deployed in the region of interest, the central node needs minimum $N \times (1 + \log_2 N + C)$ bits of memory. Where, C bits are required for keeping the sensed data of the sensor node, one bit is required for keeping the fault status as a binary decision (faulty or good) and $\log_2 N$ bits are required for keeping the sensor node's identifier.

2. For transmitting the data to the central node, each sensor node needs multi hop communication as they are far away from the central node which depletes

energy of the network quickly, especially the sensor nodes nearer to a central node.

3. The actual status of sensor node may change while central node such as sink node or base station acquire the status of the entire network in real time.

4. Here, all the sensor nodes (faulty or fault free) send their sensed data to the base station as they treat themselves as fault free before diagnosis. By doing this, the intermediate node depletes by transmitting faulty nodes data.

5. The diagnosis latency is high as it consumes time to acquire data from all the sensor nodes using multi hop communication.

6. If the central node becomes faulty, it is difficult to find the status of all sensor nodes in the network.

Distributed Diagnosis

Due to the above disadvantages of the centralized approach, the distributed fault diagnosis algorithms in WSNs [72–74] have been proposed where each sensor node participates in the diagnosis process but the final fault status is decided by the central node. Every sensor node acquires the data or output of a task from the neighboring sensor nodes and find their probable fault status by adapting the neighbor coordination, comparison, or task based approaches [75–77]. After identifying list of faulty and fault free sensor nodes, the central unit sends the status to all the participating nodes in the network. These approaches are more suitable for unconstrained based WSNs.

Wang *et al.* [78] address a distributed fault-tolerant decision fusion approach to identify sensor faults. Here, each sensor node s_i sequentially send their decisions to the base station, which needs multi hop communication. A collaborative sensor fault diagnosis (CSFD) approach is used to eliminate the unreliable local decisions when performing distributed decision fusion. Based on the pre-designed fusion rule, assuming identical local decision rules and fault-free environments, an upper bound is established on the fusion error probability. According to this error bound, a criterion is proposed to search the faulty nodes. Once the fusion center identifies the faulty nodes, all corresponding local decisions are removed from the computation of the likelihood ratios that are adopted to make the final decision.

Xu *et al.* [76] have proposed a soft fault diagnosis approach which follows a general tree to detect the faulty nodes present in the network. This paper focuses on three types of sensor faults like calibration error, random noise error, and malfunctioning. The algorithm follows three steps. The first step follows a distributed neighbor coordination based approach. In this step, each node is capable to identify its own status as either likely faulty or good by comparing its own sensed data with the neighbors data. After this step, the base station identifies any random child node to diagnose the network. If the selected node is likely good node, then it starts diagnosing process otherwise the base station again selects another child node and the process continues till a fault free child node has been selected. When a fault free node is selected, that is responsible to diagnose the entire sub tree coming under it. This node is also responsible to inform all the faulty nodes available under its sub tree to the base station. Since it detects the faulty nodes based on the general tree concept, it reduces the communication overhead and increases the network lifetime.

Andreas *et al.* [79] have proposed a Byzantine fault diagnosis method where each sensor node s_i sends a set of messages to a group of sensor nodes and also receives messages from the same group of nodes. If the number of sending and receiving messages is equal, then the sensor node is identified as fault free otherwise it is faulty. This approach needs multi hop communication and requires coordination among the nodes to identify the faulty node. Along with this it leads to congestion over the network for which the normal operation of the network is affected.

Luo *et al.* [80] have proposed a semi centralized fault diagnosis algorithm where each sensor node s_i sends a query to the sink node to know how much noise is present in its own area. The noise calculation is performed using the Bayesian and the Neyman Pearson performance criteria. After knowing own region's noise each sensor node s_i estimates the same for their neighbors Neg_i. This scheme needs multi hop communication for estimating its noise which may change during the diagnosis period.

Panda *et al.* [42] have proposed a test based semi centralized fault diagnosis approach in which the base station assigns a task to every sensor node s_i. All the sensor nodes evaluate the task and send their response to the base station. The base station analyzes the received data to identify the probable faulty sensor nodes.

The main demerits of distributed diagnosis algorithms are to allow every node to participate in the diagnosis process as a result of which there will be more message exchanges thereby lead to more energy consumption. Identifying an initiator node for the diagnosis process needs an election algorithm which is an extra computation and communication overhead for energy constraint WSN. Due to the above demerits, the researchers focus on self fault diagnosis algorithm which is explained as follows.

Self Diagnosis

Different practical applications may require the fault diagnosis to be computed in a real-time mode with a low latency, low message overhead and high throughput. Therefore, the development of self fault diagnosis approaches should aim to address these issues in addition to the aforementioned limitations of centralized and distributed approaches. Self diagnosis approaches address these issues and limitations by allowing every sensor node to keep track own and its neighboring nodes in WSNs. In these approaches, every sensor node decides independently its fault status. As a result, these approaches conserve the node energy and, consequently, prolongs the network lifetime [81]. This allows the diagnostic framework to scale for larger and denser WSNs.

2.4.10 Implementation Based

The fault diagnosis approaches can be implemented through hardware or software. Hardware based fault diagnosis approaches are suitable for the soft faulty sensor nodes as the sensor node is responsible for diagnosing its status. This is achieved by including additional hardware to the sensor node architecture for which it is cost effective, but can reduce energy overhead on the network. In software based fault diagnosis approach, predefined algorithms such as rule, estimation, time series analysis, and learning based methods are used for diagnosing the soft faulty nodes present in the network [18]. In this approach, extra software is required in the sensor node to diagnose the network.

Hardware Based

Harte *et al.* [82] have proposed a self-diagnosis architecture to monitor faulty sensor nodes present in a network. Hardware interfaces are used for fault diagnosis purpose. The hardware interface consists of a number of miniature accelerometers mounted on a flexible printed circuit board. This acts as a sensing layer around a sensor node to detect the orientation and the impact of the sensor node. It also introduces some redundancy into the design to cope with damaged accelerometers. In order to sample sensor node's reading, this design adopts several software components (e.g., ADCC, TimerC) from the TinyOS operating system.

Koushanfar *et al.* [83] have proposed self-diagnosis of sensor nodes in WSNs. This approach observes the binary outputs of its sensors by comparing with the predefined fault models. Faults caused by battery exhaustion is estimated when the hardware is competent to measure the current battery voltage [84, 85]. A diagnosis algorithm determines an estimation of the time to failure of the battery by analyzing the battery discharge curve, and the current discharge rate.

Wireless sensor node architecture is expected to be simple and energy efficient. Node self-diagnosis approach needs extra hardware which in turn increase the hardware complexity and weight of the sensor node. This approach may not be suitable for under water and body sensor network as the sensors used in those networks are light weight and less cost. These approaches do not require any message exchange among either its neighboring or surrounding nodes. Due to this reason this technique does not put any energy overhead on the sensor network and prolong the network lifetime.

The main disadvantage of the hardware based fault diagnosis approach is requirement of additional hardware cost and difficult for replacement as WSN is deployed in the human inaccessible environment. Therefore, this approach is usually not suitable for fault diagnosis in WSNs.

Software Based

In software based fault diagnosis, an algorithm is executed on one or multiple sensors in order to achieve the diagnosis of a WSN. Either independent diagnosis algorithms are developed for diagnosis purposes or diagnosis algorithm is executed with normal work load of the WSN. Yet, another way to pursue the diagnosis is based on the sensed data of normal workload, are also known as a software based diagnosis. As far as implementation is concerned, the diagnosis algorithm can be implemented either on the hardware or OS level. While hardware level implementation is suitable for terrestrial WSNs and OS level implementation are suitable in every application domain of WSNs.

2.4.11 Observation Time Based

Fault diagnosis algorithms are classified into two categories such as on line and off line based on the observed data which is collected either during the diagnosis period or prior to the diagnosis. When the data are collected during the diagnosis period, that type of diagnosis algorithms is termed as online diagnosis algorithm [6, 10, 25, 39, 40]. When the diagnosis is performed based on the previously collected and stored data of sensor nodes, the type of fault diagnosis is termed as off-line diagnosis algorithms [52, 62, 67, 68].

2.4.12 Fault Type Based

Based on the persistence of fault, the faulty sensor nodes are classified into four sub-categories such as permanent, intermittent, transient, and Byzantine faulty sensor nodes. The permanent faulty sensor nodes are identified by considering the time out mechanisms [75, 86], or minimum energy threshold [3] mechanisms. When a sensor node sends a request message to another sensor node and expects a reply within certain time duration and do not receive a reply message or the remaining energy value of a node goes below a threshold value, the sensor nodes are considered as permanently faulty. Transient faults occur once during the lifetime of a sensor node. Therefore, they are captured by checking the status of the sensor node at consecutive periods. This is more energy consuming as compared to permanent fault. The transient faulty sensor nodes are diagnosed by using any one of the fault diagnosis algorithms discussed in Section 2.4. When the sensor nodes are fault free for some duration and faulty in some other duration, the sensor node is considered as intermittently faulty. The intermittent faults are more likely in distributed systems such as multi processor and multi computer system, computer networks, wireless ad hoc networks, WSN and other kind of distributed systems.

The intermittent faulty behavior of the distributed system was first explored by Blough *et al.* [87]. Their algorithms diagnose the intermittently faulty processor by using the comparison model such as MM and MM*. As the multiprocessor systems can be powered at any time, this approach is most suitable by providing better accuracy in fault diagnosis. Bondavalli *et al.* [11] have proposed a threshold and count based intermittent fault diagnosis protocol where, they put a clear distinction between transient and intermittent faulty processor.

Khilar *et al.* [88] have presented a probabilistic based fault diagnosis approach which identifies only the intermittently faulty sensor node based on the remaining energy of the sensor node in a WSN. In their approach, each sensor node exchanges message related to their remaining energy. For this an extra message is exchanged over the network between the sensor nodes, which consumes extra energy due to message transmission and reception as a result the battery is drained quickly and lifespan of the network reduces quickly. This approach puts extra burden over the network by consuming high energy, memory and bandwidth because of the fact that the diagnosis process follows broadcast comparison model where the energy is broadcasted by each of the sensor nodes to achieve diagnosis.

Lee *et al.* [25] have presented a comparison and time redundancy matrix based fault diagnosis approach which detects both the intermittent and transient faulty sensor nodes by comparing its own sensed reading with its neighboring sensor node's data for r consecutive rounds. In each round, the sensor node collects data from their neighboring nodes and compute the absolute difference between the own sensed data with collected data and compare the result with a threshold. Here, two threshold values are used for finding the fault status of a sensor node. One threshold is used to identify partial fault status of the sensor node for each test interval and another one is to find the minimum number of times the node should declare to be faulty, so that, its final decision is to be faulty. This approach may not give good accuracy for a constant threshold. Therefore, an optimal and adaptive threshold (which changes dynamically with variation in neighboring nodes) should be designed to improve the performance of the algorithm.

To overcome the demerits as discussed earlier, Yim *et al.* [89] have proposed an adaptive and dynamically changing threshold based event diagnosis protocol to detect the events locally in the presence of intermittently faulty sensor nodes. The confidence level of the sensor node and threshold based neighbor co-ordination based approach is used for detecting the transient and intermittent faulty sensor nodes. The thresholds are adjusted dynamically to detect the events more accurately. The traditional time out mechanism is also used for detecting the permanently faulty sensor nodes.

Arunashu *et al.* [62] have proposed a hybrid fault diagnosis algorithm which diagnoses both intermittent and hard faulty sensor nodes over a static arbitrary topology network. For identifying hard faulty sensor nodes the time out mechanism is considered and neighbor co-ordination based comparison technique is used for identifying intermittently faulty sensor nodes. In time out mechanism, each node is associated with a clock value. Before the clock value, expires each node should receive some information from its neighbors. If a node is unable to receive any information from its neighbors, then the node declares that missing node as the hard faulty sensor node. In neighbor coordination based comparison technique, each sensor node compares its own sensed data with the neighbors data and the comparison is carried out over an application specific threshold value. If more than 50% of comparison result indicates that the node is faulty then that node is identified as faulty node. For calculating the time duration, number of tests required for testing the node, how many times a node should behave abnormally so that it will be declared as the faulty node. The authors put emphasis on diagnostic accuracy, diagnosis latency and energy overhead. These three parameters are modeled as the multi objective optimization problem which is solved by using multi-objective particle swarm optimization technique.

Andreas *et al.* [79] have proposed a Byzantine fault diagnosis method, where each sensor node sends a set of messages to a group of sensor nodes and also receives messages from the same group. If the number of messages sent is equal to the number of receiving messages, then the sensor node is identified as fault free otherwise faulty. This approach needs multi hop communication and requires coordination among the nodes to identify the faulty node. Recently, Kuo Feng Su *et al.* [90] presented a fault diagnosis method in WSNs where each sensor node establishes two node disjoint shortest paths [91] and send their message using this path. If the sensor node receives the same message which it had sent, then that node is identified as fault free otherwise it is labeled as faulty. This approach needs multi hop communication and requires more time to establish the path.

2.5 Conclusion

A comprehensive study of fault diagnosis algorithm is given in this chapter. It has been observed from the literature study that quite a good number of fault diagnosis schemes have been proposed for various kinds of distributed networks such as ad-hoc networks, WSNs, and wireless networks till date. The system and fault model for various kinds of systems where the diagnosis algorithms are applicable has been discussed. The classification of fault diagnosis algorithms have been presented. The suitability of self fault diagnosis algorithms have been focused as compared to centralized and distributed diagnosis which are not energy efficient. The shortcomings and advantages of various fault diagnosis algorithms are also discussed.

Chapter 3

Distributed Self Fault Diagnosis Algorithm in WSNs
using
Neighbor Co-ordination

Chapter 3

Distributed Self Fault Diagnosis Algorithm in WSNs Using Neighbor Coordination

In this chapter, a distributed self fault diagnosis algorithm is proposed to identify both hard and soft faulty sensor nodes in wireless sensor networks. The algorithm is distributed, self diagnosable and can diagnose the most common faults like stuck at zero, stuck at one, random data and hard fault. In this approach, each sensor node gathers the observed data from neighboring sensor nodes and computes the mean to check the presence of faulty sensor node which reduces the processing overhead. If a sensor node diagnoses a faulty sensor node, then it compares observed data with the data of the neighbors and predicts the probable fault status. The final fault status is determined by diffusing the fault information from the neighbors. The accuracy and completeness of the algorithm are verified based on the statistical analysis over sensors data.

3.1 Introduction

During the life span of wireless sensor networks, a number of unexpected situations arise such as the misbehavior of sensor nodes due to the occurrence of various kinds of faults [3, 9, 33, 35]. The faults occur in wireless sensor networks (WSNs) due to a number of causes such as malfunctioning of hardware and software units, malicious interference, battery exhaustion or natural calamities. The presences of faulty sensor nodes affect the performance of WSNs. This motivates us to address the issues for fault diagnosis of sensor nodes in order to obtain correct data from WSNs.

The proposed distributed self fault diagnosis algorithm considers both the soft and the hard faulty behavior of sensor nodes. In the proposed algorithm, every sensor node in the network shares their sensed data in the neighbors and predicts the probable fault status of every other sensor node. After sharing the probable fault status, the voting scheme is used as a major parameter for diagnosing the final fault status. The main contribution of this chapter includes (i) the design and evaluation of an efficient distributed self fault diagnosis algorithm for diagnosing hard and soft faulty sensor nodes in WSNs, (ii) calculate the mean to know the presence of faulty sensor node in the neighborhood, which reduces the computational time (iii) the algorithms are implemented in NS3 [38], (iv) the performance of the algorithm is compared with the existing algorithms [6,40]. The result of the proposed distributed self fault diagnosis using neighbor co-ordination approach (DSFDNC) algorithm shows that the number of communications requirement is less compared to the existing algorithms which makes the algorithm to be energy efficient.

The remaining part of the chapter is organized as follows. The system model is presented in Section 3.2. The proposed distributed self fault diagnosis algorithm using a neighbor co-ordination approach (DSFDNC) is described in Section 3.3. The algorithm has been analytically shown to be correct in Section 3.4. We described the simulation results and compared the performance with the existing fault detection algorithm in Section 3.5. Finally, Section 3.6 concludes the chapter with discussions.

3.2 System Model

The system model consists of network, fault and radio model, including the set of related assumptions. In network model, the network topology and the way sensor nodes communicate with each other are specified. In fault model, different types of fault based on faulty and fault free behavior of the sensor nodes, and the data generated by different sensor nodes are described. The radio model is used to calculate the energy required for self fault diagnosis.

3.2.1 Assumptions, Notations and Their Meanings

The proposed distributed self fault diagnosis algorithm using a neighbor coordination approach (DSFDNC) is based on the following assumptions.

1. All sensor nodes are homogeneous with uniform initial energy and transmission power.

2. Energy consumption by a sensor node is not uniform. It is because, the number of packet receptions and transmissions are not uniform due to the arbitrary network topology.

3. A sensor node works normally with the battery power of 3.3V (Ex MICAz, MRF24J40MA, CC2480A etc.).

4. Each sensor node is assigned with an Id (IP address).

5. Each sensor node sends and receives the node Id (IP address) and sensed data from their neighboring sensor nodes.

6. All the sensor nodes are static in nature in the sense that they do not change their position after deployment.

7. Links are symmetric in nature in the sense that there is a two way communication link between the sensor nodes, so that a sensor node can compute the approximate distance to another sensor node based on the received signal strength.

8. Each sensor node periodically senses the data from its immediate neighbors to diagnose its own status. The period is fixed for entire diagnosis.

9. Two neighboring sensor nodes communicate their data using UDP/IP communication protocol.

The list of notations and their meanings used in the DSFDNC algorithm are tabulated in Table 3.1.

Table 3.1: The notations used for developing the proposed DSFDNC algorithm

Symbol	**Description**
S	Set of sensor nodes in the sensor network.
s_i	A sensor node deployed at $P_i(xco_i, yco_i)$, $s_i \in S$
N	Total number of sensor nodes deployed
Neg_i	Set of neighboring sensor nodes of s_i
$CRDN_i$	Cumulative sum of received data of all neighboring nodes Neg_i of sensor node s_i
θ_1, θ_2	Threshold value used by each sensor node for detecting the status of the neighboring sensor nodes and itself
$PFFN$	Set of probable fault free sensor nodes estimated by s_i
PFN	Set of probable faulty sensor nodes estimated by s_i
RS_i	A set contains the status of s_i calculated by $s_j \in Neg_i$
$Nz(RS_i)$	Number of zero's present in the set RS_i
$No(RS_i)$	Number of one's present in the set RS_i
x_i	Modified sensed data of s_i
$MaxSense$	Maximum sensing value of the sensor node
$MinSense$	Minimum sensing value of the sensor node
$G(S, C)$	An undirected graph describing the interconnection among the sensor nodes
C	Set contains all the communication edges between the sensor nodes
T_r	Transmission range of each sensor s_i
S_1	Set of sensor nodes suffering with hard fault
S_2	Set of sensor nodes suffering with stuck at zero fault
S_3	Set of sensor nodes suffering with stuck at one fault
S_4	Set of sensor nodes suffering with random fault
S_F	Set of all faulty sensor nodes, $S_F = S_1 \cup S_2 \cup S_3 \cup s_4$
S_G	Set of fault free sensor nodes
N_i	Degree of the sensor node s_i
N_a	Average degree of sensor nodes in the network
Nx_i	A set contains received data from the neighbors Neg_i of s_i
ζ	The threshold for energy at which a sensor node s_i works normally
A	Actual sensed data of a fault free sensor node s_i
w_i	Erroneous data sensed by the sensor node s_i
Re_i	Remaining battery power of the sensor node s_i
R	Length and breadth of the terrain of interest
t	The time instant at which the data in a sensor node is observed

3.2.2 Network Model

A sensor network with N distributed sensor nodes are randomly deployed in a terrain of size $R \times R$. Each sensor node s_i, $1 \leq i \leq N$ is located in the two dimensional Euclidean plane $\mathcal{R}^2$ at $P_i(xc_i, yc_i)$, where $0 \leq xc_i, yc_i \leq R$. Sensor node s_i interacts with another set of sensor nodes $s_j \in Neg_i$ and employs a one-to-many broadcast primitive in their basic transmission mode with a single hop communication. All the sensor nodes are homogeneous and having a uniform transmission range T_r. The sensor network follows a disk model [92] where T_r of sensor node s_i is the radius of the circle centered at P_i. A sensor node s_i can interact with another sensor node s_j if the Euclidean distance $d(s_i, s_j)$ between s_i and s_j is less than or equal to T_r and otherwise, they cannot communicate with each other as defined in Equation (3.1).

$$C_{ij} = \begin{cases} 1, \ d(s_i, s_j) \leq T_r \\ 0, \ d(s_i, s_j) > T_r \end{cases} \tag{3.1}$$

The sensor network is modeled using a random graph $G(S, C)$, where S is the set of sensor nodes and C is the set of communication links between the sensor nodes. The neighboring set Neg_i (set of neighboring sensor nodes of the sensor node s_i), $Neg_i \subset S$ is defined as

$$Neg_i = \begin{cases} s_j, \ i \neq j \quad \text{and} \quad C_{ij} = 1 \\ \phi, \ i \neq j \quad \text{and} \quad C_{ij} = 0 \end{cases} \tag{3.2}$$

Here, the sensor nodes communicate with each other through an overlapping transmission range, so that most of the rectangular terrain can be covered by the deployed sensor nodes. IEEE 802.15.4 is used as the MAC layer protocol to communicate with neighboring sensor nodes. The degree of sensor node s_i is N_i which is defined as the number of one hop immediate neighbors associated with it.

An Example

Figure 3.1 depicts the arbitrary network topology based on the disk model [92]. $s_1, s_2, \ldots s_{12}$ are a set of sensor nodes, and $c_1, c_2, \ldots c_{11}$ are the communication links between the sensor nodes. A sensor node s_1 can communicate with its immediate neighbors (s_2, s_5, s_{12}) since the radius of the sensor node s_1 is within T_r. A sensor node s_1 can communicate with s_9 through its immediate neighbor s_5 which is termed as multi-hop communication. If no communication is possible for a sensor

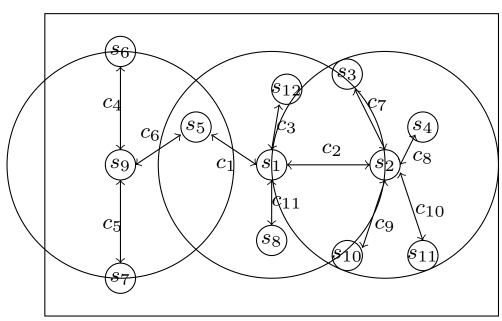


Figure 3.1: Arbitrary network topology based on disk model having $|S| = 12$ and $|C| = 10$

node with its neighbors, then this may be assumed as a hard faulty sensor node.

3.2.3 Fault Model

The arbitrary behavior of the faulty sensor nodes are classified into four subcategories such as stuck at zero, stuck at one, random and hard faulty sensor node [79]. A faulty sensor node is subjected to stuck at zero faults, if the value provided by the sensor node remains zero during identification period. When the sensor node provides maximal value (that can be the full scale value) then that type of fault is known as stuck at one. Similarly, in case of random fault, the data provided by a sensor node are arbitrary. The hard faulty sensor node remains silent throughout the life span of the network.

Let the set S_F represents the randomly chosen set of sensor nodes, which are subjected to either hard or soft fault. More specifically, let S_1, S_2, S_3, and S_4 are the set of sensor nodes suffering with hard fault, stuck at zero, stuck at one, and random fault respectively. Then, the fault free sensor nodes present in the network are $S_G = S - S_F$, where $S_F = S_1 \cup S_2 \cup S_3 \cup S_4$ and $N = |S_F| + |S_G|$.

The sensor nodes can disseminate its own sensed data to its neighbors Neg_i and also collect the observations from them at time instant t. In WSNs, some sensor nodes are subjected to a fault, whereas links are assumed to be fault free. The link faults can be detected by using error detecting and correcting codes which are usually implemented in the physical layer of the underlying networks. The fault free sensor node always provides accurate data within acceptable range, whereas faulty sensor node gives either arbitrary value in a different time or do not respond to other sensor nodes. The fault model is depicted in Figure 3.2. In Figure 3.2, 50 sensor nodes are deployed and 15 sensor nodes are faulty with 30% of fault probability which is usually assumed in WSNs.

3.2.4 Radio Model for Energy Calculation

For data communication, each sensor is equipped with a wireless transceiver. The transmitter requires transmitting electronics and amplifier whereas receiver needs only receiving electronics for data transmission. Let, α_1, α_2, and α_3 are the amount

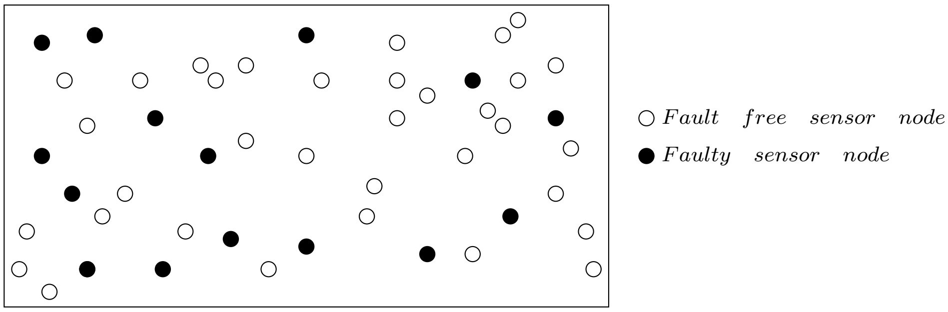


Figure 3.2: A WSN with fault free and faulty sensor nodes

of energy required for the transmitting electronics, amplifier and receiving electronics respectively. The $\alpha_1$and α_3 depends on factors such as the digital coding, and modulation, whereas the α_2 depends on the transmission distance and the acceptable bit-error rate. For data transmission and reception, the free space (fs) fading channel models are used because every sensor node needs communication to only their neighboring nodes in a single hop. Depending on the distance between the transmitter and receiver, the free space coefficient is chosen. Let, $E_T(m,d)$ and $E_R(m,d)$ are the amount of energy to transmit and receive m bytes of data over a Euclidean distance d. The total amount of energy is the sum of $E_T(m,d)$ and $E_R(m,d)$ which is given by Equation (3.3) and Equation (3.4) [93] as:

$$E_T(m,d) = m \times (\alpha_1 + \alpha_2 \times d^{\alpha}) \tag{3.3}$$

$$E_R(m,d) = m \times \alpha_3 \tag{3.4}$$

where the free space coefficient α is defined in Equation (3.5) [94].

$$\alpha = \begin{cases} 2, \ d_o \leq d \\ 4, \ d_o > d \end{cases} \tag{3.5}$$

where d_o is the minimum Euclidean distance between any two sensor nodes.

3.3 Distributed Self Fault Diagnosis Algorithm Using Neighbor Coordination

The proposed distributed self diagnosis algorithm (DSFDNC) approach has two phases such as partial self-neighbor diagnosis and voting phase. In partial self-neighbor diagnosis phase, every sensor node in the network exchanges their sensed data with the neighbors. The probable fault status of own as well as its neighbors is estimated in this phase. The estimated statuses are exchanged by all sensor nodes among themselves in voting phase. Each sensor node receives its probable fault status from the neighbors and diffuses the received status. Then each sensor node compares its computed status with diffused status to predict its own status. All the notations used for describing the steps of the DSFDNC algorithm are summarized in Table 3.1. Detail description of different phases is given below.

3.3.1 Partial Self-Neighbor Diagnosis Phase

Every sensor node $s_i \in S$ exchanges their measured data x_i with neighboring nodes Neg_i. Then, each sensor node s_i keeps the received data from the neighbors in Nx_i. After receiving the data, the partial self and neighboring node fault status are computed based on the following observations as given below.

Case 1 : The remaining battery power Re_i of sensor node s_i is computed with a constant battery power ζ to identify the hard faulty sensor node and the value for ζ is constant for all sensor nodes.

Let $MinSense$ and $MaxSense$ are the minimum and the maximum sensing value of the sensor nodes. The value of $MinSense$, and $MaxSense$ are constants and common to all the sensor nodes present in WSNs. The Case 2 and 3 are based on $MinSense$ and $MaxSense$ value and for stuck at zero and stuck at one fault as given below.

Case 2 : If the sensed data x_i of the sensor node s_i is $MinSense$, then the sensor node s_i is suffering with stuck at zero fault.

Case 3 : If the sensed data x_i of the sensor node s_i is $MaxSense$, then the sensor node s_i is suffering with stuck at one fault.

The Cases 2 and 3 are based on the fact that if the observed data of a sensor node s_i is either the value of $MinSense$ or $MaxSense$, the sensor node s_i does not depend on the neighbors to identify its own fault status. However, the Case 4 is based on the fact that if the observed data of the sensor node s_i is neither the value of $MinSense$ nor $MaxSense$, the sensor node s_i needs to find its own status as well as the neighbor's status as the observed data is random between $MinSense$ and $MaxSense$.

Case 4 : If the sensed data x_i of the sensor node s_i is between $MinSense$, and $MaxSense$, then it performs the operation defined in Equation (3.6) over the collected data from the neighboring nodes Neg_i and own sensed data x_i to identify self and neighbors probable fault status.

$$\hat{\mu}_i = \left(x_i - \frac{1}{N_i} \left(\sum_{s_j \in Neg_i} x_j \right) \right) \leq \lambda_1 \tag{3.6}$$

where λ_1 is the threshold value. When the condition given in Equation (3.6) is satisfied by sensor node s_i then include the sensor node s_i and all its neighbors $s_j \in Neg_i$ to S_G. Otherwise, the sensor node s_i and its neighboring nodes are suspected as a faulty sensor node. To identify the exact status of its own and neighboring nodes, s_i re-investigate over the received data $x_j, x_j \in Nx_i$ to identify the probable faulty sensor nodes. If the data $x_j, x_j \in Nx_i$ matched with $MinSense$, or $MaxSense$ then assign the sensor node s_j to S_2, or S_3 respectively. Otherwise, perform the following operations over the collected data Nx_i to identify the probable fault status of neighboring nodes Neg_i. The Case 4 is further partitioned into four sub cases which are given below.

Case 4(i) $|x_i - x_j| > \lambda_1$ and $x_j \leq \lambda_2$

In this case, the sensor node s_j is added to the set $PFFN_i$ and the sensor node s_i is detected as faulty sensor node.

Case 4(ii) $|x_i - x_j| > \lambda_1$ and $x_j > \lambda_2$

In this case, both the sensor nodes s_i and s_j are faulty and the sensor node s_j is added to PFN_i.

Case 4(iii) $|x_i - x_j| \leq \lambda_1$ and $x_j \leq \lambda_2$

In this case, both the sensor nodes s_i and s_j have fault free status and the sensor node s_j is added to $PFFN_i$.

Case 4(iv) $|x_i - x_j| \leq \lambda_1$ and $x_j > \lambda_2$

In this case, the sensor node s_i is fault free, the sensor node s_j is faulty and added to PFN_i.

The test outcome is 0, if a sensor node s_i is found to be fault free after performing self neighbor diagnosis phase, otherwise it is 1. After performing this, voting phase is performed as given below.

3.3.2 Voting Phase

A sensor node s_i is diagnosed as fault free if number of 0's (k) is greater than number of 1's (n), otherwise it is faulty. This leads to majority voting scheme to diagnose whether a sensor node is faulty or fault free [80]. In voting phase, each sensor node s_i exchanges its neighbor status (i.e. 0 or 1) and also receives status from its neighboring nodes Neg_i. Then predicts its own status by analyzing the status received from its neighboring nodes Neg_i i.e., each sensor node s_i counts number of 0's and 1's it has received. If number of 0's at s_i is more than number of 1's at s_i, then s_i is diagnosed as fault free and belongs to set S_G otherwise it is faulty and included in S_4 respectively. As the algorithm is self diagnosable, the Algorithm 3.1 given below is executed at each sensor node s_i to achieve distributed self fault diagnosis.

Algorithm 3.1 DSFDNC Algorithm

Data: $\mathcal{N}_\mathcal{I}$ Nodes, Nx_i
Result: Calculate S_1, S_2, S_3, S_4, and S_G
Initialize $S_1 = \phi$, $S_2 = \phi$, $S_3 = \phi$, $S_4 = \phi$, and $S_G = \phi$
Partial self-identification Phase
if $Re_i <= \zeta$ **then**
    $S_1 = S_1 \cup \{s_i\}$
else
    if $x_i = MinSense$ **then**
        $S_2 = S_2 \cup \{s_i\}$
    end
    if $x_i = MaxSense$ **then**
        $S_3 = S_3 \cup \{s_i\}$
    end
    Move to Algorithm 3.2
end
Voting Phase
$s_i \in S$ send PFN to neighbors $s_j \in Neg_i$ and receives PFN from s_j which is computed by the neighbors s_j.
From received data the sensor node s_i prepares RS_i.
if $N_z\,(RS_i) > N_o\,(RS_i)$ **then**
    Node s_i is diagnosed as fault free sensor node. $S_G = S_G \cup \{s_i\}$
else
    Node s_i is diagnosed as random faulty sensor node. $S_4 = S_4 \cup \{s_i\}$
end

Algorithm 3.2 Random fault diagnosis algorithm

```
Data: Nx_i
Result: Calculate S_G, PFN and PFFN
S_G = φ, PFN = φ and PFFN = φ
CRDN_i = 0  for j = 1···|Neg_i| and s_j ∈ Neg_i do
    CRDN_i = CRDN_i + x_j
end
CRDN_i = CRDN_i/N_i  if |x_i − CRDN_i| ≤ θ_1 then
    The node s_i and s_j ∈ Neg_i are identified as likely fault free nodes   S_G = S_G ∪ {s_i}
else
    for j = 1···|Neg_i| do
        if x_j = MinSense   or   x_j = MaxSense then
            PFFN = PFFN ∪ {s_j}
        else
            if |x_i − x_j| > θ_1 and x_j > θ_2 then
                PFFN = ∪{s_j}
            end
            if |x_i − x_j| ≤ θ_1andx_j ≤ θ_2 then
                PFN = ∪{s_j}
            end
            if |x_i − x_j| ≤ θ_1 and x_j > θ_2 then
                PFN = ∪{s_j}
            end
            if |x_i − x_j| > θ_1 and x_j ≤ θ_2 then
                PFFN = ∪{s_j}
            end
        end
    end
end
```

3.4 Analysis of the DSFDNC Algorithm

In wireless sensor networks, every sensor node s_i sense the environmental data, prepares this data into IPv6 message format and transmit to the neighboring nodes Neg_i on demand. If the sending sensor node is faulty, the actual sensed data becomes erroneous. The modified sensed data x_i of a sensor node s_i is the sum of actual sensor data A and erroneous data w_i, which is represented as Gaussian noise. It is assumed that all the sensor nodes measured same physical data and few sensor nodes (typically 5% to 30 % of sensor nodes in WSNs) can be faulty [25]. The mean of the data sensed are constant A, but the value of erroneous data differs from one sensor node to another. As the sensed value A of s_i is changed by the erroneous data w_i, the modified data x_i for each sensor node s_i follows Gaussian distribution having A mean and σ_i^2 variance, i.e. $\mathcal{N}(A, \sigma_i^2)$. It is a common assumption in WSNs literature that all the sensor node s_i measure same physical data with constant mean and different variance σ_i^2 [37].

The value of modified data x_i for s_i is given as

$$x_i = A + w_i \quad \text{where} \quad i = 1, 2, 3, \cdots, N \tag{3.7}$$

Here w_i is assumed to be independent over time and space respectively. The *probability density function*(pdf) of modified data x_i is given by [95]

$$f(x_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\frac{-(x_i-w_i)^2}{2\sigma_i^2}} \quad MinSense < x_i < MaxSense \tag{3.8}$$

The Gaussian distribution is called standard normal distribution $\Phi(x_i)$ when $w_i = 0$ and $\sigma_i^2 = 1$ and is defined in Equation (3.9).

$$\Phi(x_i) = f(x_i) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x_i^2}{2}} \quad MinSense < x_i < MaxSense \tag{3.9}$$

The probability of $x_i \in X$ $(f(x_i))$ lies in the range of $[MinSense, x_i]$ can be expressed in terms of its *cumulative distribution function*(*cdf*). As modified data $x_i \in X$ follow a Gaussian distribution, its *cdf* is defined as

$$F(x_i) = \int_{MinSense}^{x_i} f(y)dy = \Phi\left(\frac{x_i - w_i}{\sigma_i}\right) \tag{3.10}$$

Now the *cdf* can be expressed in terms of *error function*(erf). The *erf* is defined as

$$erf(x_i) = \frac{2}{\sqrt{\pi}} \int_{MinSense}^{x_i} e^{-y^2} dy \tag{3.11}$$

The *cdf* is rewritten in terms of *erf* as

$$\begin{aligned}
F(x_i) &= \int_{-\infty}^{x_i} f(y)dy \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_i} \frac{1}{\sigma_i} e^{\frac{-(y-w_i)^2}{2\sigma_i^2}} dy \\
&= \frac{1}{\sqrt{2\pi}} \left[\int_{-\infty}^{0} \frac{1}{\sigma_i} e^{\frac{-(y-w_i)^2}{2\sigma_i^2}} dy + \int_{0}^{x_i} \frac{1}{\sigma_i} e^{\frac{-(y-w_i)^2}{2\sigma_i^2}} dy \right] \\
&= \frac{1}{\sqrt{\pi}} \int_{0}^{\infty} \frac{1}{\sqrt{2}\sigma_i} e^{\frac{-(y-w_i)^2}{2\sigma_i^2}} dy + \frac{1}{\sqrt{\pi}} \int_{0}^{x_i} \frac{1}{\sqrt{2}\sigma_i} e^{\frac{-(y-w_i)^2}{2\sigma_i^2}} dy \\
&= \frac{1}{2} + \frac{1}{2} erf\left(\frac{x_i - w_i}{\sigma_i\sqrt{2}}\right) \\
F(x_i) &= \frac{1}{2}\left[1 + erf\left(\frac{x_i - w_i}{\sigma_i\sqrt{2}}\right)\right], \quad x_i, w_i \in R
\end{aligned} \tag{3.12}$$

The probability of a random variable x_i, lies in between $(\hat{\mu}_i - a)$ to $(\hat{\mu}_i + a)$ (where

$\hat{\mu}_i$ is the mean of actual data) is calculated by using its *cdf* as given below.

$$\begin{aligned} f(\hat{\mu}_i - a \le x_i \le \hat{\mu}_i + a) &= F(\hat{\mu}_i + a) - F(\hat{\mu}_i - a) \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\hat{\mu}_i + a} \frac{1}{\sigma_i} e^{\frac{-(y-\hat{\mu}_i)^2}{2\sigma_i^2}} dy - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\hat{\mu}_i - a} \frac{1}{\sigma_i} e^{\frac{-(y-\hat{\mu}_i)^2}{2\sigma_i^2}} dy \\ &= \Phi\left(\frac{\hat{\mu}_i + a - \hat{\mu}_i}{\sigma_i}\right) - \Phi\left(\frac{\hat{\mu}_i - a - \hat{\mu}_i}{\sigma_i}\right) \\ &= 2\Phi\left(\frac{a}{\sigma_i}\right) \\ &= 2\left(\frac{1}{2} erf\left(\frac{a}{\sigma_i\sqrt{2}}\right)\right) \\ &= erf\left(\frac{a}{\sigma_i\sqrt{2}}\right) \end{aligned} \tag{3.13}$$

In fact, the variance of a random variable indicates the spread of its *pdf* around the mean using Gaussian distribution, the constant a in terms of variance is chosen for better accuracy. For example, if the constant $a = 3\sigma_i$ then the probability of the random variable x_i lies in between $(\hat{\mu}_i - 3\sigma_i)$ to $(\hat{\mu}_i + 3\sigma_i)$ is

$$\begin{aligned} f(\hat{\mu}_i - 3\sigma_i \le x_i \le \hat{\mu}_i + 3\sigma_i) &= \\ erf\left(\frac{3\sigma_i}{\sigma_i\sqrt{2}}\right) \quad \text{[From (3.13)]} & \\ &= 0.9973 \end{aligned} \tag{3.14}$$

This reflects that if the variance of the erroneous data w_i at s_i is low, there is a maximum probability to get an error free measurement. If the sensor node is fault free then the variance is very low (around 0.001) [95]. The variance of error in the fault free sensor node is 1. Thus, there is a 0.9973 probability that the modified data is deviated around ±3. When the node is faulty the measured data is corrupted using normalized noise having high variance. The variance of faulty sensor node is chosen 100 times compared to fault free sensor node.

Now, we compare any two sensor nodes data x_i, and x_j at the observed time t. The difference $x_{i,j}$ is given as

$$x_{i,j} = x_j - x_i \tag{3.15}$$

Since each sensor node s_i sense the data and the error associated with sensor nodes are spatially independent, therefore x_i and x_j are independent in nature. From the definition, $x_{i,j}$ is a random value with mean $\hat{\mu}_{i,j}$ and variance $\sigma^2_{i,j}$ respectively, which are calculated as

$$\sigma^2_{i,j} = \sigma^2_i + \sigma^2_j \tag{3.16}$$

where σ_i, σ_j are the modified data variances of sensor nodes s_i and s_j respectively. When the sensor nodes are deployed in a particular environment, the sensed data for neighboring sensor nodes are nearly same. The difference is caused due to additive noise associated with the sensor data.

In general practice, for most applications of WSNs, we need the average of measured data from all sensor nodes. The theory of statistical estimation provides the mean estimator is the best *minimum variance unbiased*(MVU) estimator [95]. By considering this concept, we compared the sensor's own measured data with the mean of neighbor's data for fault identification. Let N_a be the average degree of the sensor nodes in the sensor network. The mean (θ_i) and variance (ρ^2_i) of neighbors data excluding itself is written as

$$\theta_i = \frac{1}{N_a}\sum_{j=1}^{N_a} w_j = w \quad \text{and} \quad \rho^2_i = \frac{1}{N^2_a}\sum_{j=1}^{C} \sigma^2_j \tag{3.17}$$

Now two cases arise either all neighbor nodes are fault free or some of the neighbor nodes are faulty. In the first case when all the neighboring nodes are fault free having same variance of measurement σ^2 then the mean variance is

$$\rho^2_i = \sigma^2/N_a \tag{3.18}$$

Now the difference between own measured data of sensor node s_i with the mean of its neighbors data is

$$x^m_i = x_i - \theta_i \tag{3.19}$$

which have zero mean and $\left(\frac{1+N_a}{N_a}\right)\sigma^2$ variance respectively. Let λ_1 be a constant which is used for comparing the difference such that

$$|x_i - \theta_i| \leq \lambda_1 \tag{3.20}$$

If we choose the constant $\lambda_1 = 3\left(\frac{1+N_a}{N_a}\right)\sigma^2$ for the case of all neighboring nodes are fault free with respect to itself, then there is 99.73% [from Equation (3.14)]of probability such that the absolute difference is less than λ_1.

In the second case, if any one of the neighboring node is faulty then the mean θ_i remains unchanged as all sensor nodes have same measured data. The variance of faulty sensor node is very high compared to that of fault free sensor node, ($\rho_k^2 \succ\succ \sigma^2$). Very high value for constant λ_1 cannot be chosen to satisfy the condition in Equation (3.20) because when the single neighboring node is faulty for high variation of degree the $\rho_i^2 \approx \sigma^2$. It may happen that the faulty sensor node detected as fault free. Therefore, we may lose the comparison when this comparison equation is not satisfied, then the i^{th} sensor node compares its data with the neighboring sensor nodes data using another constant λ_2. In this case, if the sensor node s_i is comparing its own data value with a faulty sensor node data value having variance σ_f^2 which is different from normal variance σ^2. Therefore the difference in variance is given as

$$\sigma_{ij}^2 = \sigma_i^2 + \sigma_j^2 = \sigma^2 + \sigma_f^2 \tag{3.21}$$

In general, the variance of faulty sensor node is nearly 100 times the variance of fault free sensor nodes. The magnitude of the difference will be compared with a higher threshold λ_2 which is $\lambda_2 = 33\sigma$.

During the comparison process, there are four different cases arise which are tabulated in Table 3.2. These four cases include either both the compared and

Table 3.2: The Comparison outcomes

Comparator	**Compared**	**comparison outcome**	$\lambda 2$
Fault free	Fault free	0	1
Fault free	Faulty	1	1
Faulty	Fault free	1	33σ
Faulty	Faulty	1	33σ

comparing sensor nodes are fault free or faulty, faulty sensor node comparing with fault free sensor node and fault free sensor node comparing with faulty sensor nodes.

When both sensor nodes are fault free, then the difference of their variance is very low, therefore, it may always satisfy with the condition for the threshold λ_2 with high probability. If both the sensor nodes are faulty with high variance, then the difference is much higher than the threshold λ_2 which indicate that one faulty sensor node can detect the status of another faulty sensor node as faulty. It is trivial when a fault free sensor node compares with the sensed value of a faulty sensor node it finds a faulty sensor node as faulty. When a faulty sensor node compares with fault free sensor node data, then the faulty sensor node makes fault free sensor node as faulty. Due to randomness of data, the results are not 100% accurate. To overcome this particular situation, we employed majority voting on the data collected from different neighboring sensor nodes before taking final decision about the fault status of a sensor node. The diagnosis latency and message complexity are computed based on the analysis given by authors Chessa *et al.* [75].

The parameters such as diagnosis latency, message complexity, storage complexity, energy consumption, network life time, completeness and correctness are considered to evaluate the performance of the proposed DSFDNC algorithm. The following lemmas i.e., Lemma 3.1 through Lemma 3.5 along with their proofs are presented below for analytical evaluation of the DSFDNC algorithm.

Lemma 3.1: The diagnosis latency of the algorithm DSFDNC is $O(2 \times T_{out} + T_{proc})$ where T_{out} is the maximum time set by the sensor node when the message exchange occurs and T_{proc} is the maximum time required by the algorithm for processing.

Proof

The diagnosis latency of the DSFDNC algorithm is the total time required to diagnose all faulty sensor nodes in the network. In the communication graph $G = V(S, C)$ of WSNs, each sensor node communicates with one hop neighboring sensor nodes only. Let T_{out} be the maximum time set by the timer when the message exchange occurs among the sensor nodes. The DSFDNC algorithm exchanges two messages, one for the sensed data and another for computing probable fault status. Therefore, the total time needed for message exchange is $2T_{out}$. Let T_{proc} be the maximum time required by the algorithm for processing both sensed data and computing fault status.

The total time required by the DSFDNC algorithm to diagnose all the faulty sensor nodes is

$$T_{DSFDNC} = O(2 \times T_{out} + T_{proc})$$

The self fault diagnosis algorithm achieves the diagnosis within a bounded delay of T_{DSFDNC}, due to synchronous WSNs as specified in the network model of Section 3.2. This proves Lemma 3.1.

Lemma 3.2: The message complexity of the DSFDNC algorithm is $O(N)$, where N is the number of sensor nodes in WSNs.

Proof

The message complexity is the total number of messages exchanged over the network to get the final fault status of all the sensor nodes in the network. The DSFDNC algorithm exchanges at most $2N$ messages for fault diagnosis.

In partial self-neighbor diagnosis phase, each sensor node s_i sends the sensed data to its neighbors, costing one message per sensor node i.e. N messages in the network. In voting phase, N number of probable fault status messages is exchanged between the sensor nodes. Therefore, the total number of messages exchanged for the DSFDNC algorithm is M_{DSFDNC} given below.

$$M_{DSFDNC} = 2N = O(N) \tag{3.22}$$

This proves Lemma 3.2.

Lemma 3.3: The storage complexity of the algorithm DSFDNC is $O(N_i \times \log_2 N)$ where N is the total number of sensor nodes present in the network, and N_i is the degree of the sensor node s_i.

Proof

In the DSFDNC algorithm, each sensor node s_i keeps the sensed data and fault status information from the neighboring nodes Neg_i in a Table NT_i. The table contains the neighbor node's ID and their sensed data. The table also contains the probable fault status of the sensor node estimated by the neighboring nodes FS_i.

The final fault status of each sensor node is estimated by looking the partial status received from the neighbors.

The node id of each sensor node s_i needs $\log_2 N$ bits. The sensed data of each sensor node is encoded using c bits (say). Similarly, the probable fault status of the neighbors estimated by each sensor node and own probable fault status received from the neighbors need 2 bits of memory. Then, the total memory required by a sensor node to keep all the required information will be $N_i(\log_2 N + 2 + c)$. For example, in a sensor network having 1024 sensor nodes where each sensor nodes data are encoded with 8 bits and the maximum degree of a sensor node assumed as 30, then the total memory requirement is $30 \times (10 + 2 + 8) = 600$ bits or 75 Bytes. The total storage required for achieving the diagnosis is $O(N_i \times \log_2 N)$. This proves Lemma 3.3.

Lemma 3.4: The total energy required to achieve diagnosis by DSFDNC algorithm is $\sum_{i=1}^{N} E_i(m + p, T_r)$ where N is the total number of sensor nodes present in the network and E_i is the total energy consumption by the sensor node s_i, m is the message size during data exchange, p is the message size during status exchange, and T_r is the maximum distance between any two sensor nodes.

Proof

The energy requirement of the network to detect the soft faulty sensor node by using the DSFDNC algorithm is calculated. A sensor node consumes energy for data transmission and processing. Since processing required less energy (because of the development of low power VLSI and computing architecture), the energy required for data transmission is considered here. The DSFDNC algorithm needs message exchange twice by each sensor node. The energy calculation for each message transmission is provided separately.

A. The energy required for exchanging the sensed data

Let $E_1, E_2, \ldots, E_N$ be the amount of energy dissipated by the sensor nodes $s_1, s_2, \ldots, s_N$ respectively. Let c be the message size of sense data and T_r (transmission range) be the maximum distance a sensor node can transmit the message. Thus the total amount of energy required by a sensor node s_i for transmission of c

bits of message (data) is

$$E_{Ti}(m, T_r) = m \times [\alpha_1 + \alpha_2 \times T_r^{\alpha}] \tag{3.23}$$

where α_1, α_2, and α are the constants defined in the radio model.

This transmission energy is common for all the senor nodes in the network. The energy required to receive the data from various neighboring nodes are different, because of variance in the degree of the sensor nodes. The energy required by a sensor node s_i is to receive data from all the neighboring nodes is given as

$$E_{Ri}(c, T_r) = N_i \times c \times \alpha_3 \tag{3.24}$$

where N_i is the degree of sensor node s_i and α_3 is the constant defined in the radio model discussed in Section 3.2.

Therefore, the total amount of energy required by the sensor node s_i for data transmission and reception is

$$E1_i(c, T_r) = E_{Ti}(c, T_r) + E_{Ri}(c, T_r) \tag{3.25}$$

B.The Energy required for exchanging the probable fault status

Each sensor node s_i exchanges p bits of information (fault status of its neighbors) to its neighbors. Following the same procedure discussed above, the total energy required by the sensor node s_i here is given as

$$E2_i(p, T_r) = E_{Ti}(p, T_r) + E_{Ri}(p, T_r) \tag{3.26}$$

Where $E_{Ti}(p, T_r) = p \times [\alpha_1 + \alpha_2 \times T_r^{\alpha}]$ and $E_{Ri}(p, T_r) = N_i \times p \times \alpha_3$.
Finally, the total energy required for each sensor node to diagnose soft faulty sensor nodes in the network is given as

$$\begin{aligned} E_i(c + p, T_r) &= E1_i(c, T_r) + E2_i(p, T_r) \\ &= (c + p) \times (\alpha_1 + \alpha_2 \times T_r^{\alpha} + N_i \times \alpha_3) \end{aligned}$$

The total energy consumed by the network of N sensor nodes for identifying the faulty sensor node is

$$E_{total}(c,d) = \sum_{i=1}^{N} E_i(c+p, T_r) \tag{3.27}$$

This proves Lemma 3.4.

Table 3.3: Comparison of proposed scheme over the existing algorithms

Parameters	**DSFDNC Algorithm**	**DFD Algorithm [6]**	**IDFD Algorithm [40]**	**Lee *et al.* [25]**
Number of message exchanges	$2N$	$5N$	$3N$	kN
Diagnosis latency	$2T_{out} + T_{proc}$	$5T_{out} + T_{proc}$	$3T_{out} + T_{proc}$	$kT_{out} + T_{proc}$
Energy	$(m+p)N(\alpha_1 + \alpha_2 d^{\alpha} + d_i\alpha_3)$	$(2m+3)N(\alpha_1 + \alpha_2 d^{\alpha} + d_i\alpha_3)$	$(2m+1)N(\alpha_1 + \alpha_2 d^{\alpha} + d_i\alpha_3)$	$kmN(\alpha_1 + \alpha_2 d^{\alpha} + d_i\alpha_3)$
Memory Requirement	$N_i log_2 N + 2 + c$	$2N_i log_2 N + 3 + c$	$2N_i log_2 N + 2 + c$	$kN_i log_2 N + k + c$

Lemma 3.5: The proposed DSFDNC algorithm is correct and complete.

Proof

According to the diagnosis literature [96, 97], an algorithm is said to be complete, if no sensor node remains undiagnosed after the diagnosis process over. An algorithm is said to be correct if a faulty sensor node is diagnosed as faulty with better diagnosis accuracy which is defined in Section 3.5.

In order to prove the completeness property, we consider the parameters such as transmission range, the average degree of the network and mean of the observed data of the different neighboring sensor nodes. The Algorithm DSFDNC performs the diagnosis on each sensor node of the sensor network. As the WSN is a connected network, a sensor node gets at least one immediate neighbor sensor node which is coming within it's transmission range. The degree of the WSN is at least one. The proposed diagnosis algorithm runs on each sensor node to achieve its status based on the observed and estimated data of its neighbor nodes. In fact, the algorithm does not need the data from the sensor nodes beyond a sensor node's one-hop neighbors in order to achieve diagnosis. Since every sensor node participates in the diagnosis process, all the sensor nodes achieve diagnosis using neighbor coordination approach which satisfies the completeness property of the diagnosis algorithm.

To prove the correctness of the proposed DSFDNC algorithm, we consider the parameters diagnosis accuracy, false alarm rate, and false positive rate. If a fault free sensor node is wrongly diagnosed as faulty and faulty sensor node is wrongly diagnosed as fault free, it is difficult to find the correct status of a sensor node. However, the correctness of the proposed DSFDNC algorithm is ensured with high diagnosis accuracy, less false positive rate and less false alarm rate which are optimal results based on Gaussian distribution within a range of minimum and maximum sensing values of each sensor node in WSNs. This proves Lemma 3.5.

3.5 Simulation Model

In this section, the proposed DSFDNC algorithm is implemented using the network simulator NS3 [38] and the performances are compared with the existing algorithms proposed by authors Jiang (IDFD algorithm) [40] and Chen *et al.* (DFD algorithm) [6]. An arbitrary topology is created by considering $N = 512$ number of sensor nodes with average degree $5, 9, 16, 21$, and 25. The distance between sensor nodes are set according to range propagation loss model. The network parameters used for evaluating the algorithms are given in Table 3.4. The faults such as hard fault,

Table 3.4: Simulation parameters

Parameter	Value
Network size	512 sensor nodes
Average degree	5, 9, 16, 21
Topologies	Arbitrary network
Propagation loss model	Range propagation loss model
MAC	IEEE 802.15.4
Simulation time	300s
Fault model	Normal random variable
α_1	50 nJ/bit
α_2	10 pJ/bit/m^2
α_3	50 nJ/bit
T_r	$(35, 40, 54, 60)m$
Network grid	From $(0, 0)$ to $(500, 500)m$
λ_1	3
λ_2	33
Initial energy	1J

stuck at zero, stuck at one, and random fault are introduced into the network. It is assumed that the various faults are independent of each other. The performance of the algorithms is evaluated in terms of diagnosis accuracy, false alarm rate, false positive rate, and number of message exchanges, energy consumption, diagnosis latency, and network life time. These parameters are in Section 1.1.4.

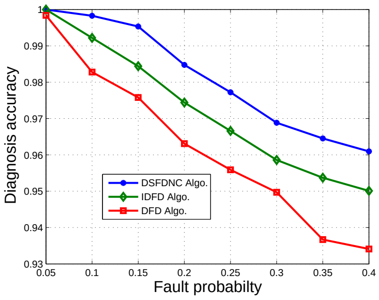
3.5.1 Results and Discussion

The performance of the DSFDNC algorithm is analyzed and compared with existing algorithms for different fault probabilities (P_f), the average degrees (N_a) of sensor nodes in the network and the predefined threshold values (λ_1 and λ_2). After random deployment of the sensor nodes in a rectangular terrain of size 500×500, an arbitrary network topology is formed and any sensor node sends the data within its transmission range T_r. The performances are measured by varying the fault probabilities from 0.05 to 0.4 with step size of 0.05. The threshold values λ_1 and λ_2 used in DSFDNC algorithm are taken as 3 and 33 respectively.

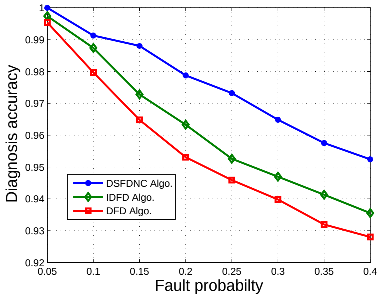
In the simulation model, the data of a fault free sensor node are generated by using *normal distribution* function with mean $A = 30$ and variance $\sigma^2 = 1$. The faulty sensor nodes are assumed to have the same mean as fault free node, but the variance is chosen 100.

3.5.2 Performance of the Algorithm with Respect to diagnosis accuracy, false positive rate and false alarm rate

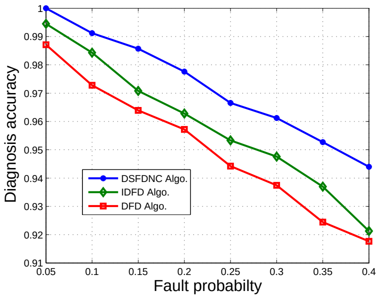
The diagnosis accuracy, false positive rate and false alarm rate versus the fault probabilities for different N_a are plotted in Figure 3.3(a) to Figure 3.3(d), Figure 3.4(a) to Figure 3.4(d) and Figure 3.5(a) to Figure 3.5(d) respectively. As we can see from Figure 3.3, Figure 3.4 and Figure 3.5, the diagnosis accuracy, false alarm rate, and false positive rate of the proposed DSFDNC algorithm is improved as compared to that of existing DFD [6] and IDFD [40] algorithm by 2% and 1% respectively. This improvement in diagnosis accuracy, false positive rate, and false alarm rate of the proposed algorithm over the DFD and IDFD algorithms is due to the statistical property of the mean, which is used for comparison of own fault status with neighbors data. Each sensor node does not take its own fault decision by only comparing own data with one of the neighbors. Instead the fault status is found by each of the neighbor sensor nodes. Then a voting scheme among the probable fault status measured by the neighboring nodes is used to take the final decision. That helps each sensor node to take correct decision about the fault status.



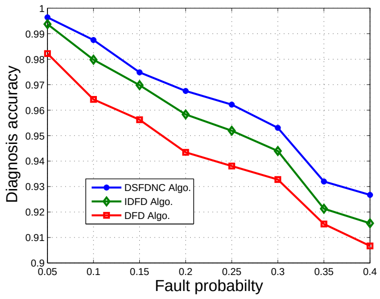
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$

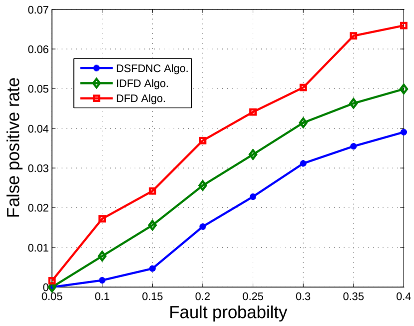


(c) For average degree $N_a = 16$

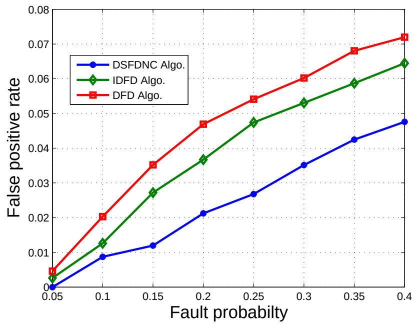


(d) For average degree $N_a = 21$

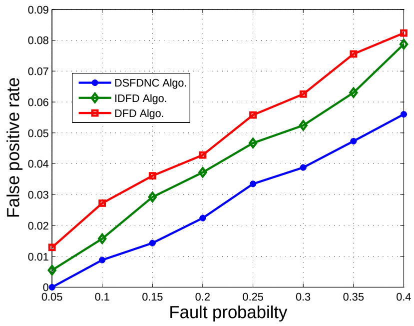
Figure 3.3: Diagnosis accuracy versus fault probability plots for the DSFDNC, DFD and IDFD algorithms.



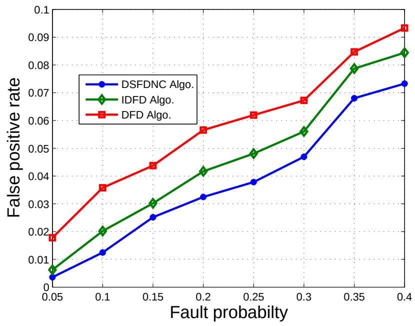
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$

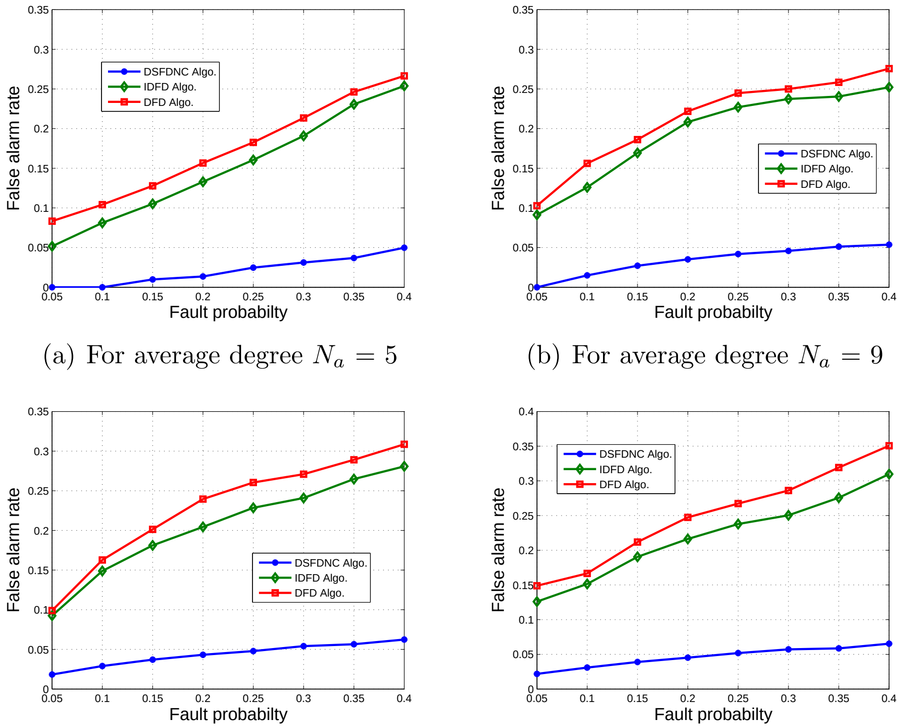


(c) For average degree $N_a = 16$



(d) For average degree $N_a = 21$

Figure 3.4: False positive rate versus fault probability plots for the DSFDNC, DFD and IDFD algorithms.



(a) For average degree $N_a = 5$

(b) For average degree $N_a = 9$

(c) For average degree $N_a = 16$

(d) For average degree $N_a = 21$

Figure 3.5: False alarm rate versus fault probability plots for the DSFDNC, DFD and IDFD algorithms.

Ideally the DSFDNC algorithm aims to achieve the diagnosis accuracy is to be 1 and false alarm rate should be 0. In Figure 3.3, Figure 3.4 and Figure 3.5, the proposed algorithm attains these ideal performance for lower fault probability; this degrades for higher fault probabilities. In the worst case scenario (40% of fault probability with average degree 25) the diagnosis accuracy, false positive rate and false alarm rate of the DSFDNC algorithm are 0.92, 0.08 and 0.05 respectively. The DFD and IDFD algorithms diagnosis accuracy, false positive rate and false alarm rate performances are around (0.88, 0.9), (0.09,0.11) and (0.33, 0.36) respectively. Therefore, in the worst case scenario, the proposed diagnosis DSFDNC algorithm gives improvement of 4% in diagnosis accuracy, 3% in false positive rate, and 3% in false alarm rate over DFD and IDFD algorithms respectively.

3.5.3 Message Complexity

Total number of messages exchanged in DFD, IDFD and proposed DSFDNC algorithms depend on the number of sensor nodes present in WSNs. As the message exchange is the only means of diagnosis, each sensor node has to exchange the diagnostic message with their neighboring nodes in order to achieve the diagnosis. Therefore, the message complexity is independent of fault probability and the average degree of the network. In fact, every sensor node participates in the diagnosis process.

The proposed DSFDNC algorithm has resulted in 33% and 60% less message exchange overhead as compared to that of IDFD and DFD algorithms. This is due to the requirement of multiple messages from the neighboring nodes. The DSFDNC algorithm needs two messages from the neighboring nodes to diagnose the faulty sensor node, in the worst case, DFD and IDFD algorithms need 5 and 3 messages respectively to identify the status of the faulty sensor node in WSNs.

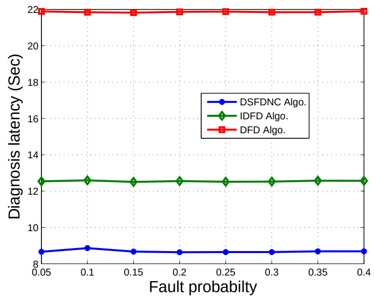
Table 3.5 shows the comparison of the number of messages required by existing and proposed DSFDNC algorithm. As every message bit transmission and reception consumes some amount of energy which is more than a bit computation at the sensor node, the proposed DSFDNC algorithm requires only two types of messages (i.e. sensed data and status information), this leads to less energy consumption also, and therefore energy efficient.

Table 3.5: Total number of messages exchanged for DSFDNC, DFD, and IDFD algorithms

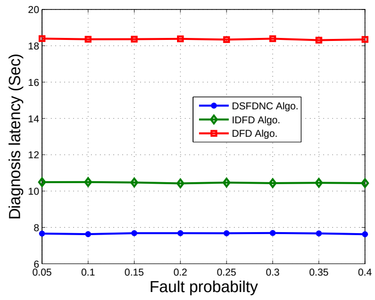
Algorithm	DSFDNC Algorithm	DFD Algorithm	IDFD Algorithm
$N_a = 10$	1024	2560	1536
$N_a = 15$	1024	2560	1536
$N_a = 20$	1024	2560	1536

3.5.4 Diagnosis Latency

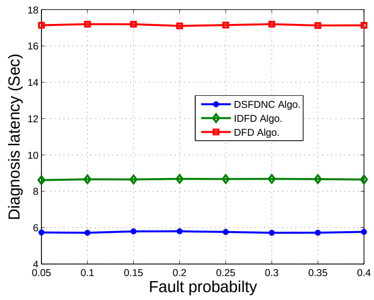
The diagnosis latency is used for evaluating the DSFDNC algorithm which measures the time required to diagnose all the faulty sensor nodes in WSNs. The diagnosis latency versus fault probability of the DSFDNC, DFD, and IDFD algorithms for different average degrees are depicted in Figure 3.6. From the figure it is shown that there is 54% and 33% improvement of diagnosis latency in the new algorithm as compared to that of the DFD and IDFD algorithms. The DSFDNC algorithm is also scalable due to the fact that there is no change in diagnosis latency with respect



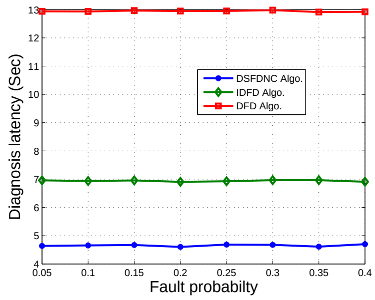
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$



(c) For average degree $N_a = 16$



(d) For average degree $N_a = 21$

Figure 3.6: Diagnosis latency versus fault probability for the DSFDNC, DFD, and IDFD algorithms.

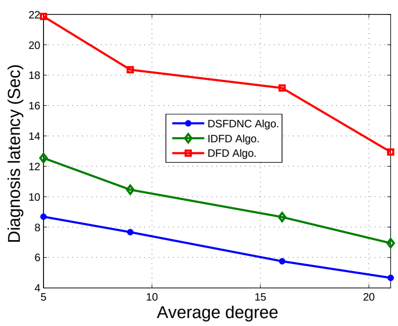
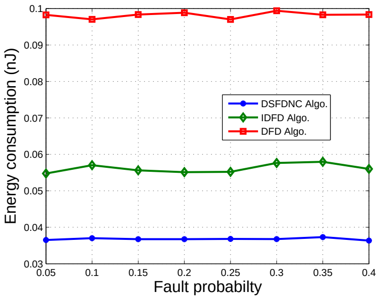
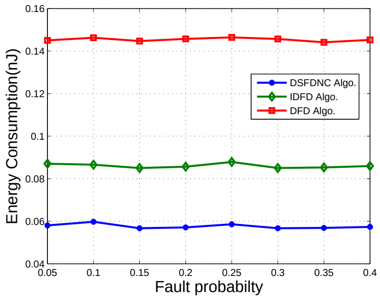


Figure 3.7: Diagnosis latency versus average degree for the DSFDNC, DFD and IDFD algorithms

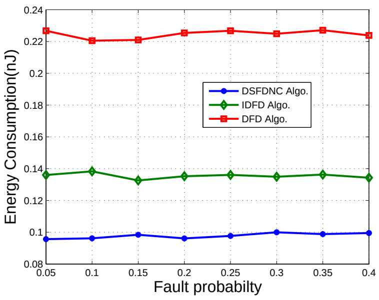
to increasing fault probabilities in WSNs. The diagnosis latency with respect to varying network average degrees is depicted in Figure 3.7.



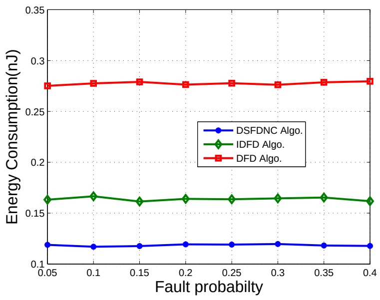
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$



(c) For average degree $N_a = 16$



(d) For average degree $N_a = 21$

Figure 3.8: Total energy consumption versus fault probability for the DSFDNC, DFD, and IDFD algorithms.

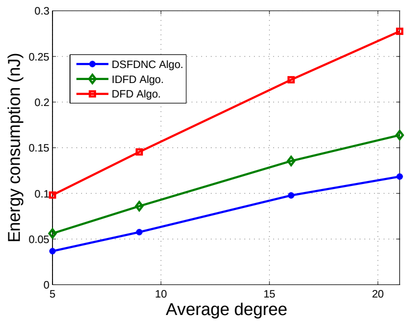
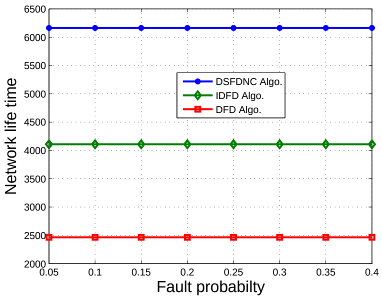


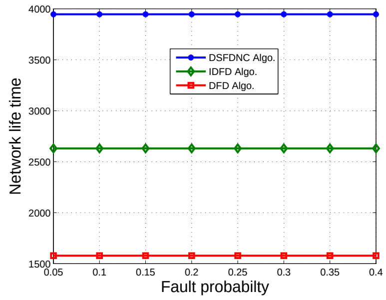
Figure 3.9: Total energy consumption versus average degree for the DSFDNC, DFD and IDFD algorithms

3.5.5 Energy Consumption

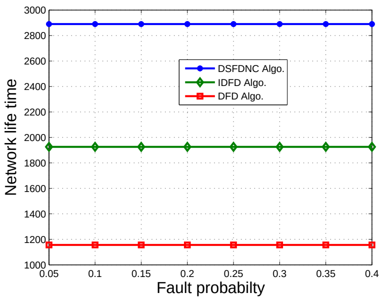
Figure 3.8 depicts the total energy consumed in the network for fault diagnosis by the DSFDNC, DFD and IDFD algorithms for different fault probabilities. The result shows that as N_a increases, the energy consumption increases. The energy consumption in the DSFDNC is 28% and 56% less consumed as compared to that of IDFD and DFD algorithms. The number of message receptions is varied due to packet loss in the network for a fixed number of message transmissions. As more energy is required for message transmission than reception, the DSFDNC requires less number of messages for transmission and thereby consume less energy compared to the existing algorithms. The energy consumption with respect to varying N_a are depicted in Figure 3.9. It is noted that the DSFDNC does not use any special message for diagnosis rather the message containing observed data of the sensor nodes are utilized.



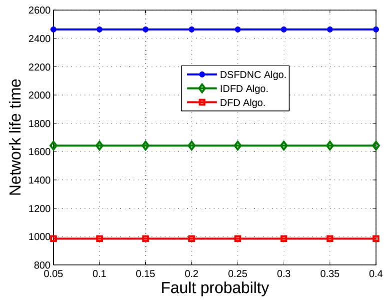
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$



(c) For average degree $N_a = 16$



(d) For average degree $N_a = 21$

Figure 3.10: Network life time versus fault probability for the DSFDNC, DFD, and IDFD algorithms.

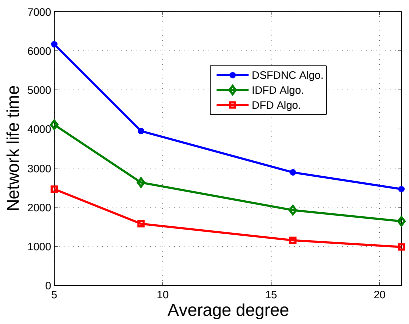


Figure 3.11: Network life time versus average degree for the DSFDNC, DFD and IDFD algorithms

3.5.6 Network Life Time

The network life time of the DSFDNC, DFD and IDFD algorithms with respect to varying N_a and f_p is shown in Figure 3.11 and Figure 3.10 respectively. From the figures, it is found that the network life time for DSFDNC algorithm is 33% and 66% less as compared to that of IDFD and DFD algorithms respectively. This is due to the DSFDNC needs less energy compared to the IDFD and DFD algorithms. The network life time decreases with the increase in average degrees of sensor nodes in WSNs.

Improvement in the results of the DSFDNC over DFD and IDFD algorithms is tabulated in Table 3.6 for $N_a = 16$ and fault probability $P_f = 0.3$.

Table 3.6: Improvement of DSFDNC algorithm over DFD and IDFD algorithms when $N_a = 16$ and $P_f = 0.3$

Performance parameter	DSFDNC algorithm	DFD Algorithm	IDFD Algorithm	Improvements over DFD Algorithm	Improvements over IDFD Algorithm
Diagnosis accuracy	0.943976	0.911297	0.92766	3%	2%
False alarm rate	0.0523	0.3086	0.2807	3%	2%
False positive rate	0.0561	0.0888	0.0724	3%	2%
Message exchange	1024	2560	1536	60%	33%
Network life time	2890	1156	1926	60%	33%
Eenergy consumption	0.0995251	0.223813	0.134288	55%	28%
Diagnosis latency	5.76698	17.1334	8.6442	54%	33%

3.6 Conclusion

The distributed self fault diagnosis algorithm using neighbor coordination (DSFDNC) is proposed in this chapter based on a realistic fault model such as stuck at zero, stuck at one, random and hard fault. The accuracy and completeness of the DSFDNC algorithm are evaluated by using the neighbor coordination method. The result shows that the diagnosis accuracy and false positive rate of the new algorithm is improved by 3%, and 1% as compared to that of DFD and IDFD algorithms when the average degree of the network is 15. The algorithm outperforms over the DFD and IDFD algorithms by providing higher network life time and lower diagnosis latency due to less consumption of energy and message overhead on WSNs. In order to improve the performance of the DSFDNC algorithm, in the forthcoming chapter, we use the hypothesis testing based approach to diagnose the soft faulty sensor node instead of comparing the observation with the mean of their neighbors data.

Chapter 4

Distributed Self Fault Diagnosis Algorithm in WSNs *using* Hypothesis Testing

Chapter 4

Distributed Self Fault Diagnosis Algorithm for WSNs Using Hypothesis Testing

The existing fault diagnosis algorithms in wireless sensor networks based on comparison of neighbor node's data require more computation and communication overheads and yields poor performance when the degree of the network is less. This chapter presents a novel distributed fault diagnosis algorithm to diagnose soft faulty sensor nodes by gathering information from the neighbors. The developed scheme is based on the Newman-Pearson test to predict the fault status of each sensor node. The performance is evaluated in terms of diagnosis accuracy and false alarm rate. The simulation results show that the performance of the proposed algorithm is much better when the average degree of sensor nodes is less. The time and message complexity, diagnosis latency, network life time and energy consumption of the algorithm are also analyzed.

4.1 Introduction

Distributed self fault diagnosis in wireless sensor networks (WSNs) have been the main focus of research in recent years [5,16,33,35]. This is due to the fact that, the sensor nodes are deployed in human inaccessible and hostile environments, where the sensor nodes are subjected to hard and soft faults. In fact, soft faults are more frequent than the hard faults [16]. The occurrence of these faults in sensor nodes prevents the normal operation of the WSNs in various ways. In WSNs, the accuracy of the observed data is sent by a sensor node is important for the overall network's performance. Therefore, diagnosis of soft faulty sensor node (the sensor nodes which accumulates erroneous readings) is an essential issue of the reliability of WSNs [33, 35].

In Chapter 3, a distributed self fault diagnosis algorithm based on neighbor coordination is developed where the sensor nodes are comparing the data with the mean of neighbors data. Since the mean approaches to its true value if the number of samples is more (central limit theorem), it needs more number of neighboring nodes. In this chapter, a distributed self fault diagnosis algorithm is developed which can provide better diagnosis accuracy for lower average degree network. Instead of comparing own data with the mean of neighbor's data here the statistical hypothesis testing is chosen to diagnose the faulty sensor node. Further, in order to minimize computation and communication in the fault diagnosis process, each sensor node first tests the presence of faulty sensor node in its neighbor and then predicts the probable fault status of each of them. For this, the Neyman-Pearson (NP) detection method is used. Then, each sensor node shared the probable fault status among the neighbors. A fusion scheme is used at each of the sensor nodes to take the final decision on its fault status as discussed in Chapter 3.

The major contribution of this chapter are (i) the design and evaluation of an efficient distributed self fault diagnosis algorithm using hypothesis testing (DSFDHT) for diagnosing soft faulty sensor nodes in WSNs, (ii) the Neyman-Pearson (NP) detection method is used to diagnose the faulty sensor node (iii) the performance is compared with the existing distributed algorithms such as DFD [6] and IDFD [40], (iv) the algorithms are implemented in NS3 [38]. (v) The performance of the algorithms is evaluated using generic parameters as discussed in Chapter 3.

The remaining part of the chapter is organized as follows. The system models assumed for the proposed algorithm DSFDHT are provided in Section 4.2. The proposed distributed self fault diagnosis algorithm is described in Section 4.3. The analysis of the DSFDHT algorithm and its correctness is given in Section 4.3.2. The simulation results are provided in Section 4.4. Finally, Section 4.5, concludes the chapter with a discussion.

4.2 System Model

The system model for this work is similar to that of Chapter 3 except the fault model, where only soft fault is considered in this chapter. It is because the soft faults are more frequent in WSNs and diagnosing those soft faults are more challenging than hard faults.

4.2.1 Assumption, Notation and Meaning

In addition to the assumptions made in Chapter 3, the following assumptions are considered for the DSFDHT algorithm.

1. The sensor nodes are subjected to permanent, stuck at zero, stuck at one, and random faults.

2. The communication links are assumed to be fault free.

The list of notations and their meanings used in the DSFDHT algorithm are tabulated in Table 4.1.

4.2.2 Network and Radio Model

In this chapter, we consider the network and radio model same as specified in Chapter 3, assuming that the deployment scenario of WSNs remains same i.e., a large class WSNs in human inaccessible and hostile environments.

4.2.3 Data and Fault Model

It is assumed that all the fault free sensor nodes in WSNs are measuring the same physical value at any given instant of time t and some of the sensor nodes may be faulty. The data of the sensor node s_i at time instant t, denoted as $x_i(t)$ is generated from the normal Gaussian distribution with mean $\hat{\mu}$ and variance σ^2 i.e., $x_i(t) \approx N(\hat{\mu}, \sigma^2)$. The data model assumes that all the fault free sensor nodes should have the same mean $\hat{\mu}$ and the error in the sensed data of different sensor nodes is

Table 4.1: The notations used for developing the proposed DSFDHT algorithm

Symbol	**Description**
S	Set of sensor nodes in WSNs.
s_i	A sensor node deployed at $P_i(xco_i, yco_i)$
N	Number of sensor nodes deployed
P_r	Probability of faulty sensor node in WSNs
S_1	Set of sensor nodes suffering with stuck at zero fault in WSNs, $S_1 \subset S$.
S_2	Set of sensor nodes suffering with stuck at one fault in WSNs, $S_2 \subset S$.
S_3	Set of sensor nodes suffering with random faults in WSNs, $S_3 \subset S$.
N_F	Number of faulty sensor nodes deployed, $N_F < N$
S_F	Set of faulty sensor nodes in WSNs, $S_F \subset S$.
S_G	Set of fault free sensor nodes in WSNs, $S_G \subset S$.
NT_i	Neighboring table of s_i containing all the information about its neighbors and itself.
FS_i	Fault status of the sensor node s_i
Neg_i	A set containing all the neighboring sensor nodes of s_i
$\mathcal{N}_f^i$	A set containing all the faulty sensor nodes in Neg_i, $\mathcal{N}_f^i \subset Neg_i$
$FSNeg_{i,j}$	Fault status of the neighboring sensor node s_j estimated by s_i
$CRDN_i$	Cumulative sum of receiving data of all neighbors of sensor node s_i
γ_1, γ_2	Threshold value used by each sensor node s_i for estimating the status of the neighboring sensor nodes and itself
$PFFN_i$	Probable fault free sensor node estimated by the sensor node s_i
PFN_i	Probable faulty sensor node estimated by sensor node s_i
RS_i	A set containing status of s_i calculated by $s_j \in Neg_i$
N_i	Degree of the sensor node $s_i \in S$
$Nz(RS_i)$	Number of zero's in the set RS_i
$No(RS_i)$	Number of one's in the set RS_i
$x_i(t)$	Sensed data of ith sensor node at time t
$MaxSense$	Maximum sensing value of the sensor node s_i
$MinSense$	Minimum sensing value of the sensor node s_i
$G(S, C)$	An undirected graph describing the interconnection among the sensor nodes to form an arbitrary network topology
C	Set containing all the communication edges between the sensor nodes in S
T_r	Transmission range of sensor nodes
μ	Mean of the sensor node's measurement data which is assumed to be constant for all sensor nodes.

different. In fact, the data model having same mean $\hat{\mu}$ and different variance σ^2 is followed for the fault diagnosis in WSNs in the diagnosis literature [37].

We consider the set S_F of sensor nodes are subjected to failure. Three types of soft faulty sensor nodes are considered such as stuck at zero, stuck at one, and random fault [79]. The aim of the proposed DSFDHT algorithm is to diagnose such faulty sensor nodes in sparse WSNs. Let S_1 , S_2 and S_3 are the set of randomly chosen sensor nodes suffering with stuck at zero, stuck at one, and random fault. The fault free sensor nodes in the network are $S_G = S - S_F$, where $S_F = S_1 \cup S_2 \cup S_3$ and $|S - S_F| >> |S_1 \cup S_2 \cup S_3|$ and $N = S_G \cup S_F$.

The sensor node can disseminate its own sensed data to its neighbors Neg_i and also collect the observation $\{x_j\}_{s_j \in Neg_i}$ from neighbors. It stores the neighbors data in local memory for further use.

4.3 Distributed Self Fault Diagnosis Algorithm using Hypothesis Testing (DSFDHT)

4.3.1 Description of the Algorithm

The proposed DSFDHT algorithm is divided into three phases such as (i) fault diagnosis phase (ii) fault status exchange phase and (iii) decision phase. The details about each phase are given as follows.

(i) **Fault diagnosis phase**

In this phase, each sensor node s_i estimates the status of the neighboring nodes Neg_i from the received data Nx_i on a round basis. The fault diagnosis is based on the binary hypothesis testing. Let h_{ij} is the binary decision (0 or 1) taken by the sensor node s_i of the neighboring node s_j, where $s_j \in Neg_i$. $h_{ij} = 0$ if the sensor node s_i decides the hypothesis H_0 (for fault free sensor node) otherwise, $h_{ij} = 1$ if the hypothesis H_1 (for faulty sensor node). In order to minimize the computation involved in fault finding, this phase is further divided into two steps. In the first step, the sensor node s_i diagnose the presence of faulty sensor node in the neighborhood. If the sensor node s_i finds the presence of faulty sensor node based on the estimation, then it evaluates the second step to search the exact faulty sensor node.

In the first step, to find either faulty sensor node is present in the neighbor or not, each sensor node s_i estimates the mean of the data received Nx_i from the neighboring nodes Neg_i. The mean $\hat{\mu}_i$ estimated for sensor node s_i is given by Equation (4.1) as:

$$\hat{\mu}_i = \frac{1}{N_i + 1}\left(x_i + \sum_{s_j \in Neg_i} x_j\right) \tag{4.1}$$

Then, each sensor node s_i performs the following operations.

$$|\mu - \hat{\mu}_i| \leq \gamma_1 \tag{4.2}$$

Where μ is the actual sensed value and γ_1 is the threshold value which common to all sensor nodes. When the condition given in Equation (4.2) is satisfied by sensor node s_i, the status of all the neighbors $s_j \in Neg_i$ (including its own) set is concluded as a probable fault free sensor node. In the same Step 1, we define $h_{ii} = 0$, $h_{ij} = 0$, and $s_j \in Neg_i$. Otherwise, the sensor node re-investigate over the received data $\{x_j\}_{s_j \in Neg_i}$ to identify the exact faulty sensor node.

Step 2 is further categorized into the following steps.

(i) If the received data x_j is $MinSense$ or $MaxSense$ then assign the fault status FS_j as one and include the sensor node s_j into the set S_1 or S_2 respectively.

(ii) If x_j is any random value other than zero or maximum then perform the following operation

$$|\mu - x_j| \leq \gamma_2 \tag{4.3}$$

where γ_2 is another constant which is common to all the sensor nodes. If the condition given in Equation (4.3) is satisfied, then the sensor node s_j is likely fault free for s_i. The fault status is set as $h_{ij} = 0$ and added to $PFFN_i$. Otherwise, s_j is identified as likely faulty and added to PFN_i by changing the fault status value $h_{ij} = 1$.

In this way, each sensor node s_i predicts its own probable fault status as well as the status of the neighbors $s_j \in Neg_i$. The value of the thresholds γ_1 and γ_2 depend on the network and the probability of faulty sensor nodes which is discussed in Section 4.3.2.

(ii) **Fault status exchange phase**

In this phase, all the sensor nodes exchange their predicted fault status $FsNeg_{ij}$ of the neighbors to themselves. Finally, each sensor node s_i receives a fault status of its own, predicted by its neighboring sensor nodes Neg_i. In this phase, only a single bit of diagnosis information about the fault status of different sensor nodes is used in order to reduce the message size.

(iii) **Decision phase**

In this phase, each sensor node s_i has own fault status decided by the neighboring nodes. The sensor node s_i makes the final decision after fusing all the possible decision received from the neighboring nodes Neg_i. The optimum fusion rule is a

k-out-of-n rule is used to calculate own fault status [80].

Let $h_i = 0$ if the sensor node s_i finally decides H_0 and $h_i = 1$ if the sensor node s_i decides H_1 after fusing the received status as per the following conditions given in Equation (4.4) as

$$h_i = \begin{cases} 1, \ \sum_{j=1}^{N_k} h_{ji} \geq k \\ 0, \ \sum_{j=1}^{N_k} h_{ji} < k \end{cases} \tag{4.4}$$

where k is an integer between 1 to N_k and i varies from $1, 2, ..., N_i$ for sensor node s_i.

Since the algorithm is distributed, it runs by the individual sensor node. The detailed algorithm for self fault diagnosis is discussed in Algorithm 4.1.

4.3.2 Analysis of the DSFDHT Algorithm

In this section, the proposed DSFDHT algorithm has been analyzed to prove its correctness. In WSNs, every sensor node s_i sense the environment by using an appropriate sensor (for example temperature or pressure) and then transmits to the neighboring nodes Neg_i as per the requirements. While performing this, the error is likely to be added with the sensed data during measurement time. As per the data model discussed in Section 4.2.3, the model for the fault free sensor node's measured data is given as

$$x_i(t) = \mu + w_i(t) \quad \text{where} \quad s_i \in S_G \tag{4.5}$$

Where $x_i(t)$ is the measured data of sensor node s_i at t^{th} instant, μ is the mean of sensor node's measurement (actual sensed data) and $w_i(t)$ is the additive Gaussian error at s_i having zero mean $\hat{\mu}_i$ and variance σ_i^2. $x_i(t)$ and $w_i(t)$ are assumed to be independent over time and space respectively.

The *probability density function*(pdf) of $x_i(t)$ is given as [95]

$$f(x_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\frac{-(x_i-\mu)^2}{2\sigma_i^2}} \tag{4.6}$$

When the sensor node becomes faulty, the data is modeled as error only with the same variance as that of a fault free sensor node, which is given as

$$x_i(t) = w_i(t) \quad : \text{faulty sensor node} \quad \text{where} \quad s_i \in S_F \tag{4.7}$$

Then each sensor node s_i broadcast the data $x_i(t)$ to the neighbors as per the network topology and also accumulates data from the neighbors Neg_i. Now, two situations arise. Either all the sensor nodes including it are fault free or any one and/or more than one sensor nodes are faulty.

$$\begin{aligned} H_0 : x_i(t) &= \mu + w_i(t) \quad \text{where} \quad s_i \in Neg_i \quad \text{and} \quad s_i \notin \mathcal{N}_f^i \\ H_1 : x_i(t) &= w_i(t) \quad \text{where} \quad s_i \in \mathcal{N}_f^i \end{aligned} \tag{4.8}$$

where H_0 and H_1 are the two hypothesis tests for fault free and faulty sensor node respectively. The modified data $y_i(t)$ is estimated as follows.

$$y_i(t) = \mu - x_i(t) \tag{4.9}$$

Now, the modified data for the hypothesis H_0 and H_1 becomes

$$\begin{aligned} H_0 : y_i(t) &= w_i(t) \quad \text{where} \quad i \in Neg_i \quad \text{and} \quad s_i \notin \mathcal{N}_f^i \\ H_1 : y_i(t) &= \mu + w_i(t) \quad \text{where} \quad s_i \in \mathcal{N}_f^i \end{aligned} \tag{4.10}$$

Since the error is symmetrically distributed around zero mean, we can write $w_i(t)$ instead of $-w_i(t)$. Out of N_i sensor nodes only $N_f(\prec N_i)$ sensor nodes are assumed to be faulty. Therefore $\overline{y}$ is computed as given in Equation (4.11).

$$\overline{y} = \frac{1}{N_i} \sum_{s_j \in Neg_i} y_j(t) \tag{4.11}$$

The variance of $\overline{y}$ is $\frac{\sigma^2}{N_i}$. and

$$\begin{aligned} E[\overline{y}] &= E\left[\frac{1}{N_i} \sum_{s_j \in Neg_i} (\mu - x_j(t))\right] \\ &= \frac{1}{N_i} E\left[\sum_{s_j \in Neg_i \& s_j \notin \mathcal{N}_f^i} (\mu - x_j(t)) + \sum_{s_j \in \mathcal{N}_f^i} (\mu - x_j(t))\right] \\ &= \mu_{eff} \end{aligned} \tag{4.12}$$

where

$$\mu_{eff} = \frac{N_f^i * \mu}{N_i} = P_r\mu \tag{4.13}$$

where $\mathcal{N}_f^i$ is the set of faulty neighbor nodes and N_f^i is the number of faulty sensor nodes in the neighbor of sensor node s_i. Now the NP detector decides H_1 if

$$\begin{aligned} &\tfrac{P(y_i:H_1)}{p(y_i:H_0)} > \nu \\ \Rightarrow &\frac{\frac{1}{(2\pi\sigma^2)^{N/2}}\exp\left[-\frac{1}{2\sigma^2}\sum_{i\in Neg_i}(y_i(t)-\mu_{eff})^2\right]}{\frac{1}{(2\pi\sigma^2)^{N/2}}\exp\left[-\frac{1}{2\sigma^2}\sum_{i\in Neg_i}(y_i^2(t))^2\right]} > \nu \end{aligned} \tag{4.14}$$

Taking logarithm on both sides and simplifying, the results are given as

$$\frac{1}{N_i}\sum_{i=1}^{N_i} y_i(t) > \frac{\sigma^2}{N_i * \mu_{eff}}\ln\nu + \frac{\mu_{eff}}{2} = \nu' \tag{4.15}$$

Since the μ_{eff} is unknown because it depends on the number of faulty sensor nodes in the neighboring of s_i, so ν' cannot be evaluated. Clearly it shows that the sample mean does not depend on μ_{eff} but the threshold ν' does. To overcome this dependence, use the definition of the probability of false alarm rate (PFA). Under H_0, $\overline{y} \sim \mathcal{N}(0, \sigma^2/N)$. Hence

$$\begin{aligned} PFA = Pr(\overline{y}_i > \nu'; H_0) \qquad &= Q\left(\frac{\nu'}{\sqrt{\frac{\sigma^2}{N}}}\right) \\ \Rightarrow \nu' \quad &= \sqrt{\frac{\sigma^2}{N}}Q^{-1}(PFA) \end{aligned} \tag{4.16}$$

Which is independent of μ_{eff}. Since the pdf of $\overline{y}$ under H_0 does not depend on μ_{eff}, the threshold which is chosen to maintain a constant pdf, can be found and will not depend on μ_{eff}. However, that probability of diagnosis accuracy P_D will depend on the value of μ_{eff}. More specifically

$$P_D = P_r(\overline{y}_i > \nu'; H_1) \tag{4.17}$$

and

$$\overline{y}_i \sim \mathcal{N}(\mu_{eff}, \frac{\sigma^2}{N}) \tag{4.18}$$

so that

$$P_D = Q\left(\frac{\nu' - \mu_{eff}}{\sqrt{\frac{\sigma^2}{N}}}\right) \tag{4.19}$$

As expected, P_D increases with increasing μ_{eff}. Further, from Equation (4.19), it is clearly shown that the P_D depends upon ν' and σ^2. The value of ν' is chosen from the given PFA as in Equation (4.16) and σ^2 is calculated from the given SNR.

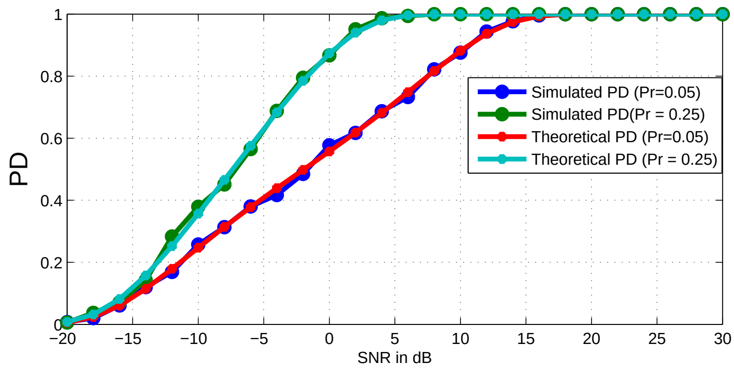


Figure 4.1: Theoretical Equation (4.19) and simulated plots for SNR versus PD

The first step in the proposed self fault diagnosis algorithm is to test whether any faulty sensor node is in the neighbor or not by using Equation (4.16) and Equation (4.19). This minimizes the amount of computation and communication in the fault finding process. Since each sensor node has to check the presence of faulty sensor node by gathering information from neighbors, the number of sensor nodes $N = 15$ (equal to average degree) is chosen for analysis. The data for each sensor node is generated by using the model given in Equation (4.5). The variance of the error σ^2 of a fault free sensor node is calculated according to the given SNR. To find the presence of faulty sensor node in the neighborhood, we followed the steps stated above and implemented in MATLAB [98]. Repeat the experiment 1000 times and the average result is plotted in Figure 4.1 between SNR Vs P_D for different probability of faults (5 and 25 percent). From the figure, the $SNR = 20$ dB is chosen so that the probability of diagnosis $P_D = 1$ for a particular value of $PFA = 10^{-2}$.

In the second step of the proposed algorithm, we compare the measured data with the true value. Each sensor node compares $y_i = \mu - x_i$ for all the received data from the neighbors based on Equation (4.20) as defined below.

$$\begin{aligned} H_0 &: y_i = w_i \quad \text{where} \quad s_i \in Neg_i \\ H_1 &: y_i = \mu + w_i \quad \text{where} \quad s_i \in Neg_i \end{aligned} \tag{4.20}$$

Then, using NP test, H_1 will be decided if

$$\begin{aligned} \frac{P(y_i; H_1)}{P(y_i; H_0)} &= \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i-\mu)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-y_i^2}{2\sigma^2}\right)} > \nu_2 \\ &\Rightarrow \exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2} + \frac{y_i^2}{2\sigma^2}\right) > \nu_2 \end{aligned} \tag{4.21}$$

Taking logarithm on both sides and simplifying we get

$$y_i > \frac{\sigma^2}{\mu} * \ln \nu_2 + \frac{\mu^2}{2 * \sigma^2} = \nu_2' \tag{4.22}$$

where

$$\nu_2' = \frac{\sigma^2}{\mu} \ln \nu_2 + \frac{\mu^2}{2 * \sigma^2} \tag{4.23}$$

Now choose the value of ν_2 such that P_D is maximum for a given value of PFA.

$$\begin{aligned} PFA_i = &\quad P_r(y_i > \nu_2'; H_0) \\ = &\quad \int_{\nu_2'}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \left(e^{-t^2}\right) dt = Q\left(\frac{\nu_2'}{\sigma}\right) \end{aligned} \tag{4.24}$$

And the probability of diagnosis is given as

$$P_{D_i} = P_r(y_i > \nu'; H_1) \tag{4.25}$$

$$P_{D_i} = Q\left(\frac{\nu_2' - \mu}{\sigma}\right) \tag{4.26}$$

Based on the above binary hypothesis testing, each sensor node takes probable fault decision about the neighboring node and transmits to the neighboring node. Each sensor node has the own fault status decided by the neighboring nodes. A sensor node takes the final decision after fusing all the fault status received from the neighbors. The optimum fusion rule is a k-out-of-n rule used [99].

The derivation of different parameters such as diagnosis accuracy, false alarm rate, false positive rate, message complexity, storage complexity, energy consumption, diagnosis latency, and network life time remains same as given in Chapter 3. In order to find the message complexity, storage complexity, energy consumption, diagnosis latency, and network life time for the proposed algorithm DSFDHT, the Lemma 4.1 to Lemma 4.4 are used. The lemmas are as follows.

Lemma 4.1: The diagnosis latency of the DSFDHT algorithm is $O(2 \times T_{out} + T_{proc})$ where T_{out} is the maximum time set by the timer when the message exchange occurs among the sensor nodes and T_{proc} is the maximum time required by the algorithm for processing.

Proof

The diagnosis latency of DSFDHT algorithm is same as that of DSFDNC algorithm as discussed in Lemma 3.1 of Chapter 3.

Lemma 4.2: The message complexity of the DSFDHT algorithm is $O(N)$ where N is the total number of sensor nodes in WSNs.

Proof

The message complexity of DSFDHT algorithm is same as that of DSFDNC algorithm as discussed in Lemma 3.2 of Chapter 3.

Lemma 4.3: The storage complexity of the DSFDHT algorithm is $N_i \times (\log_2 N + 2 + c)$ where N is the total number of sensor nodes in the network, N_i is the degree of the sensor node s_i and c is the constant value required for encoding the sensed data.

Proof

The memory requirement of DSFDHT algorithm is same as that of DSFDNC algorithm as discussed in Lemma 3.3 of Chapter 3.

Lemma 4.4: The energy complexity of the DSFDNC algorithm is $\sum_{i=1}^{N} E_i(m+p, T_r)$ where N is the total number of sensor nodes in the network and $E_i(m+p, T_r)$ is the total energy consumption by s_i.

Proof

The energy consumption of DSFDHT algorithm is same as that of DSFDNC algorithm as discussed in Lemma 3.4 of Chapter 3.

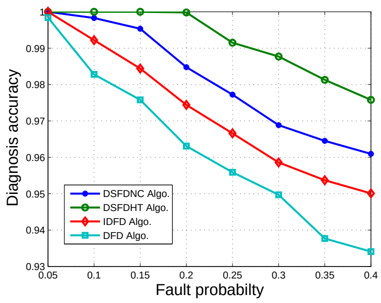
Algorithm 4.1 DSFDHT Algorithm

Data: N_I Nodes, Nx_i
Result: Calculate S_1, S_2, S_3, and S_G
Initialize $S_1 = \phi$, $S_2 = \phi$, $S_3 = \phi$, and $S_G = \phi$
Fault-diagnosis Phase
if $x_i = MinSense$ **then**
  $S_1 = S_1 \cup \{s_i\}$
end
if $x_i = MaxSense$ **then**
  $S_2 = S_2 \cup \{s_i\}$
else
  $X_i = x_i$ **for** $j = 1 \cdots |Neg_i|$ *and* $s_j \in Neg_i$ **do**
    $X_i = X_i + x_j$
  end
  $X_i = X_i/(N_i + 1)$ **if** $|\mu - X_i| \leq \gamma_1$ **then**
    $S_G = S_G \cup \{s_i\}$ **for** $j = 1 \cdots |Neg_i|$ *and* $s_j \in Neg_i$ **do**
      $S_G = S_G \cup \{s_j\}$
    end
  else
    $PFFN = \phi$ $PFN = \phi$ **for** $j = 1 \cdots |Neg_i|$ **do**
      if $x_j = MinSense$ *or* $x_j = MaxSense$ **then**
        $PFN = PFN \cup \{s_j\}$
      else
        if $|\mu - x_j| \leq \gamma_2$ **then**
          $PFFN = PFFN \cup \{s_j\}$
        else
          $PFN = PFN \cup \{s_j\}$
        end
      end
    end
  end
end
Fault status exchange phase
$s_i \in S$ send PFN to neighbors $s_j \in Neg_i$ and receives PFN from s_j which is computed by the neighbors s_j.
From received data the sensor node s_i prepares RS_i.
Decision phase
if $N_z(RS_i) > N_o(RS_i)$ **then**
  Node s_i is diagnosed as fault free sensor node. $S_G = S_G \cup s_i$
else
  Node s_i is diagnosed as random faulty sensor node. $S_3 = S_3 \cup \{s_i\}$
end

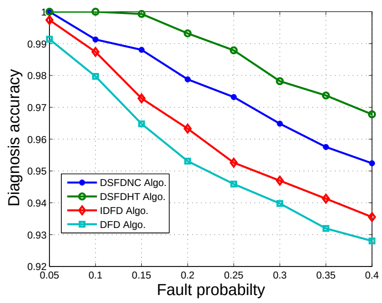
4.4 Simulation Model

This section provides examples to illustrate the advantages of using the Newman Pearson test for self fault diagnosis and to examine the performance of the proposed DSFDHT algorithm against the previously described DSFDNC, DFD and IDFD algorithms in Chapter 3. The simulation is carried out using discrete event simulation library NS3 (Network Simulation version 3.19) [38]. The performances of the algorithms are evaluated in terms of diagnosis accuracy, false alarm rate, false positive rate, number of message exchanges, energy consumption, diagnosis latency, and network life time. The simulation parameters used in NS3 simulator are provided in Table 3.4 of Chapter 3.

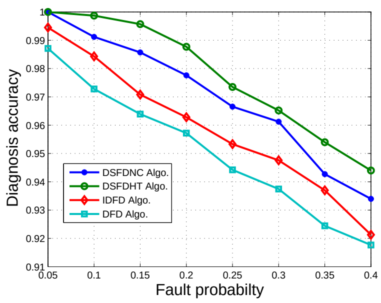
The algorithms are tested for different fault probabilities of the sensor nodes from



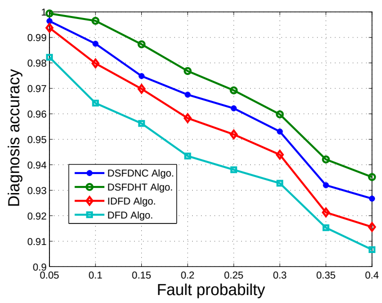
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$

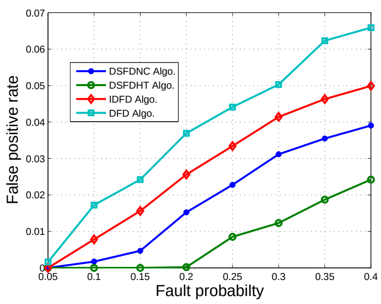


(c) For average degree $N_a = 16$

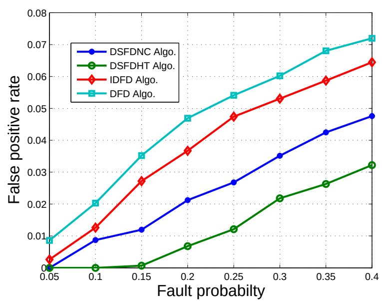


(d) For average degree $N_a = 21$

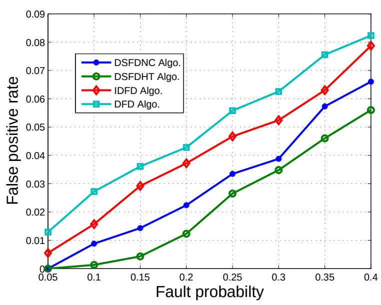
Figure 4.2: Diagnosis accuracy versus fault probability plots for the DSFDHT, DSFDNC, DFD and IDFD algorithms.



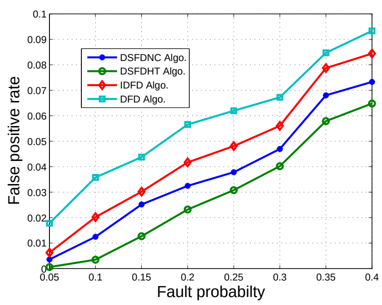
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$

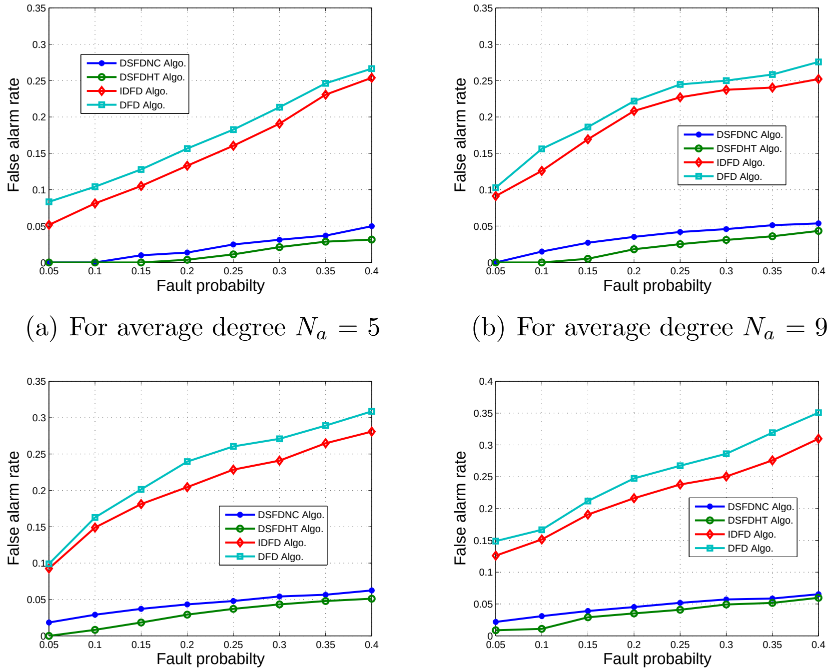


(c) For average degree $N_a = 16$



(d) For average degree $N_a = 21$

Figure 4.3: False positive rate versus fault probability plots for the DSFDHT, DSFDNC, DFD and IDFD algorithms.



(a) For average degree $N_a = 5$ (b) For average degree $N_a = 9$

(c) For average degree $N_a = 16$ (d) For average degree $N_a = 21$

Figure 4.4: False alarm rate versus fault probability plots for the DSFDHT, DSFDNC, DFD and IDFD algorithms.

0.05 to 0.4 in the step size of 0.05. The statistical NP detector's performance depends on the number of data and to study the behavior of the algorithm in both sparse and dense network, the proposed DSFDHT algorithm is verified for the average degrees 5, 9, 16, and 21. In order to get the average degrees of sensor nodes as 5, 9, 16, and 21 in the network, the transmission ranges are chosen 35, 40, 54, and 60 respectively. We have performed 100 experiments for each point of the plot. It has been seen from the Figure 4.2 to Figure 4.4 that, the performance of the proposed DSFDHT algorithm is better compared to that of the DSFDNC, DFD and IDFD algorithms.

The diagnosis accuracy, false positive rate and false alarm rate for different average degrees and fault probabilities are plotted in Figure 4.2(a) to Figure 4.2(d), Figure 4.3(a) to Figure 4.3(d) and Figure 4.4(a) to Figure 4.4(d) respectively. As we can see from Figure 4.2, Figure 4.3 and Figure 4.4, the DSFDHT algorithm yields 2%, 4% and 6% more diagnosis accuracy, 1%, 3% and 5% more false positive rate and 2%, 15% and 20% more false alarm rate over DSFDNC, DFD and IDFD algorithm respectively. The accurate fault diagnosis is observed in DSFDHT algorithm because, (1) the efficient hypothesis detector criterion is used by each sensor node to detect the fault status, (2) each sensor node estimate the fault status of its own as well as the neighbor nodes, and (3) fusion scheme is used for final fault status decision. On the other hand, the existing algorithms produce a less accurate estimate of fault status because of the simple comparison model is used for fault diagnosis.

For all the algorithms, the accuracy of diagnosis decreases when number of faulty sensor nodes in the network increases. The result shows that the lower average degree of sensor nodes of 5, 9 or 16, the DSFDHT algorithm's performance is significantly better compared to that of the DSFDNC, DFD, IDFD algorithms. In fact, the less average network uses a less number of communications which makes the algorithm energy efficient. In the worst case scenario, the diagnosis accuracy is not less than 95% for the DSFDHT algorithm, whereas the DFD and IDFD algorithms provide diagnosis accuracy of 90% and 92% respectively. When the network has less than 15% faulty sensor nodes, the DSFDHT algorithm diagnoses the faulty sensor nodes with diagnosis accuracy of 100%, where as the existing algorithms provide less than 100% diagnosis accuracy.

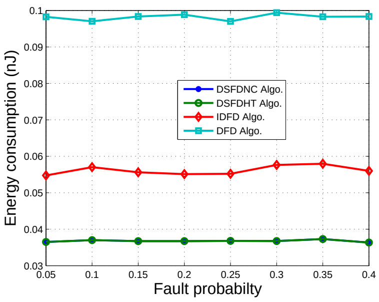
In Figure 4.4, the false alarm rate performance of the DSFDHT algorithm outperforms over other DSFDNC, DFD, and IDFD algorithms. In the worst case, the false alarm rate does not exceed 6%, where as the DFD and IDFD algorithms produce near about 40% of false alarm rate. This shows that the algorithm has greater potential to diagnose a fault free sensor node as fault free.

4.4.1 Message Complexity

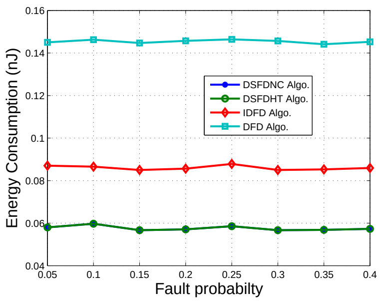
The message complexity of the DSFDHT algorithm is illustrated here. The number of message exchanges required in an algorithm depends on network size i.e., number of sensor nodes. The DFD [6] and IDFD [40] algorithms require more message overhead as compared to the DSFDHT algorithm. In the worst case, the DFD and IDFD algorithms requires 5 and 3 number of message exchanges over the network to identify the faulty status of the sensor nodes in the network whereas DSFDHT and DSFDNC approach needs only 2 messages. Usually the message complexity is independent of fault probability, because in soft fault diagnosis method, it is assumed that all the sensor nodes communicate with their neighbors by using single hop communication. In Table 4.2, total number of messages exchanged by different algorithms for different average degrees are presented. The proposed DSFDHT algorithm requires 33% and 60% less message exchange overhead as compared to that of the IDFD and DFD algorithm, whereas the number of message exchanges for the DSFDHT and previously described DSFDNC approach remains same.

Table 4.2: Total number of messages exchanged for the DSFDHT, DSFDNC, DFD, and IDFD algorithms

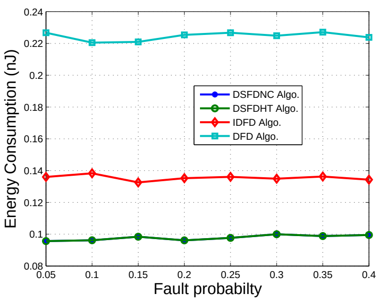
Average degree (N_a)	DSFDHT Algorithm	DSFDNC Algorithm	DFD Algorithm	IDFD Algorithm
5	1024	1024	2560	1536
9	1024	1024	2560	1536
16	1024	1024	2560	1536
21	1024	1024	2560	1536



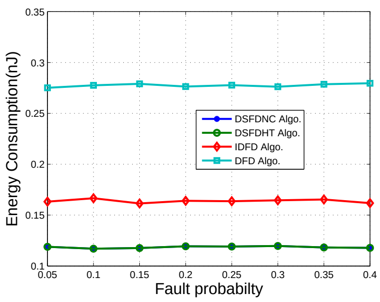
(a) For average degree $N_a = 5$



(b) For average degree $N_a = 9$



(c) For average degree $N_a = 16$

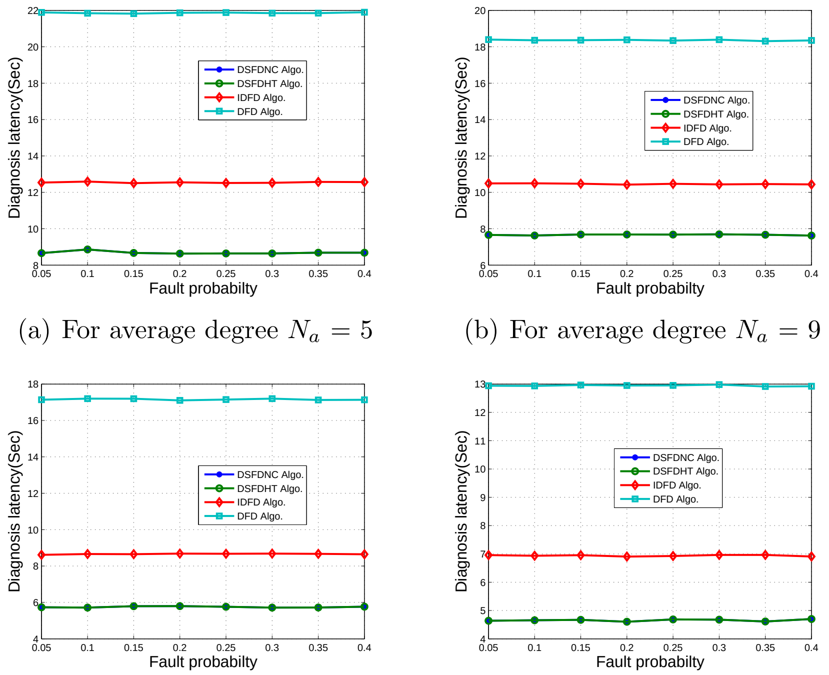


(d) For average degree $N_a = 21$

Figure 4.5: Total energy consumption versus fault probability for the DSFDHT, DSFDNC, DFD, and IDFD algorithms.

4.4.2 Energy Consumption

Figure 4.5 depicts the total energy used in the network for fault finding by the DSFDHT, DFD and IDFD algorithms for different fault probabilities. Energy is used for both message transmission and reception. Since less number of messages is required for the DSFDHT algorithm to detect the faulty sensor node, thus the



(a) For average degree $N_a = 5$

(b) For average degree $N_a = 9$

(c) For average degree $N_a = 16$

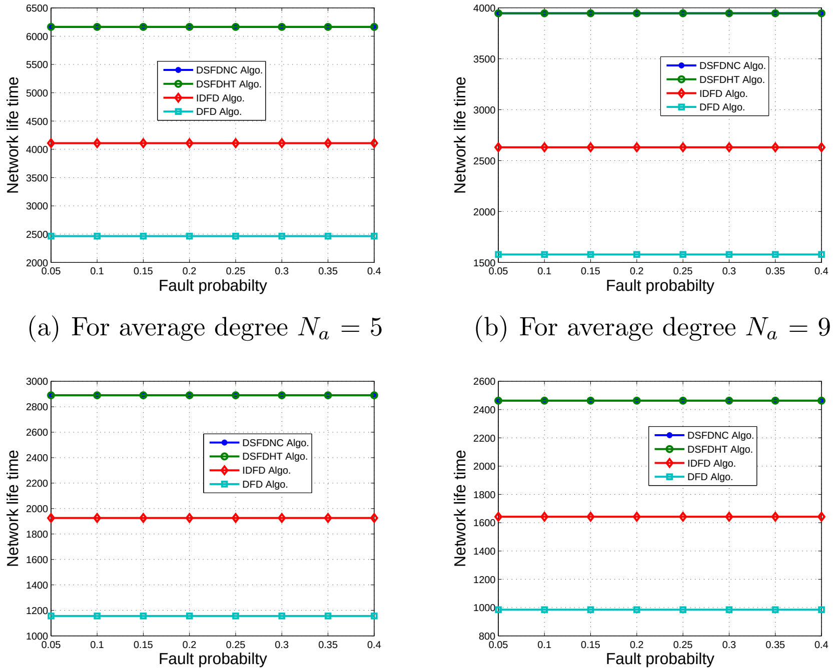
(d) For average degree $N_a = 21$

Figure 4.6: Diagnosis latency versus fault probability for the DSFDHT, DSFDNC, DFD, and IDFD algorithms.

algorithm needs less energy. The number of message receptions is varied due to packet loss in the network for a fixed number of message transmissions. As we know that, more energy is required for transmission than reception, the DSFDHT requires less energy compared to other algorithms. Along with this, the energy consumption by varying network average degrees is depicted in Figure 4.9.

4.4.3 Diagnosis Latency

The diagnosis latency is one of the generic parameter for self fault diagnosis algorithms available in literature [6, 40]. The diagnosis latency of the DSFDHT, DFD and IDFD algorithms with respect to varying average degrees of the sensor nodes and fault probabilities are plotted in Figure 4.6 and Figure 4.8 respectively. It shows that the algorithm DSFDHT has 50% and 33% less diagnosis latency as compared to that of DFD and IDFD algorithms and remains same as that of DSFDNC algorithm. It is because the diagnosis latency depends on the number of message exchanges in the network and the DSFDHT algorithm has reduced one and two numbers of messages necessary to achieve the diagnosis as compared to that of IDFD and DFD



(a) For average degree $N_a = 5$

(b) For average degree $N_a = 9$

(c) For average degree $N_a = 16$

(d) For average degree $N_a = 21$

Figure 4.7: Network life time versus fault probability for the DSFDHT, DSFDNC, DFD, and IDFD algorithms.

algorithms respectively. It remains same for different fault probabilities as depicted in Figure 4.6. The diagnosis latency with respect to varying network average degrees are depicted in Figure 4.8 which shows that as the average degree increases, the diagnosis latency decreases. This is due to the fact that as the average degree of the network increases the sensor nodes are coming closer to each other.

4.4.4 Network Life Time

The network life time of the DSFDHT, DSFDNC, DFD and IDFD algorithms with respect to varying fault probabilities and average degrees are shown in Figure 4.10 and Figure 4.7. From the figure it is found that the lifetime for DSFDHT algorithm is same as that of DSFDNC algorithm and 58%, and 34% more compared to that of DFD and IDFD algorithms. It is because, the DSFDHT algorithm needs the same number of message exchanges as that of DSFDNC and one, two numbers of less message exchanges as compared to that of IDFD and DFD algorithms. From the Figure 4.7, it is found that the network life time decreases with the increase in average degrees of sensor nodes in the network. This is because the number of

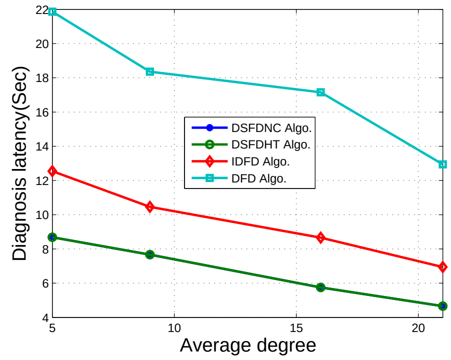


Figure 4.8: Diagnosis latency versus average degree N_a for the DSFDHT, DSFDNC, DFD and IDFD algorithms

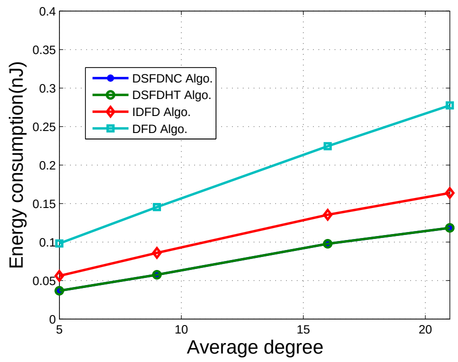


Figure 4.9: Total energy consumption versus average degree N_a for the DSFDHT, DSFDNC, DFD and IDFD algorithms

communications are required more if the degrees of the network increases.

Performance improvement of the DSFDHT over DSFDNC, DFD and IDFD algorithms are given in Table 4.3 for $N_a = 16$ and fault probability $P_f = 0.3$.

4.5 Conclusion

A self-diagnosable distributed fault diagnosis algorithm is proposed for the diagnosis of soft faulty sensor nodes in WSNs. The diagnosis algorithm is based on the NP detection method on a realistic fault model. The accuracy and completeness are analyzed by assuming the sensed data which is mixed with random noise. The algorithm is implemented in NS3 and compared to the performance of other existing algorithms. From the simulation, it is evident that the performance of the DSFDHT algorithm is better in terms of diagnosis accuracy, false positive rate, and false alarm rate as compared to that of DSFDNC algorithm proposed in Chapter 3. The

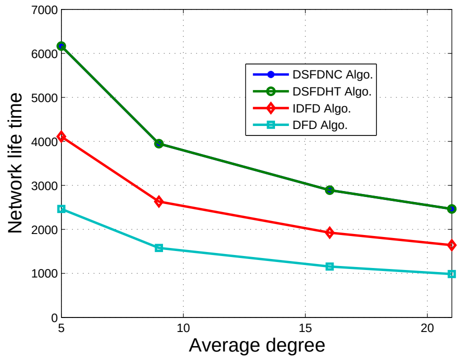


Figure 4.10: Network life time versus average degree N_a for the DSFDHT, DSFDNC, DFD and IDFD algorithms

Table 4.3: Performance comparison of the DSFDHT over DSFDNC, DFD and IDFD algorithms when $N_a = 16$ and $P_f = 0.3$

Performance parameter	DSFDHT Algorithm	DSFDNC Algorithm	DFD Algorithm	IDFD Algorithm	Improvements over DSFDNC Algorithm	Improvements over DFD Algorithm	Improvements over IDFD Algorithm
Diagnosis accuracy	0.9652	0.941216	0.93759	0.92744	2%	3%	4%
False alarm rate	0.0432	0.0641247	0.240833	0.270833	2%	20%	23%
False positive rate	0.0348	0.0588	0.0625	0.0725	2%	3%	4%
Message exchange	1024	1024	2560	1536	0%	60%	33%
Network life time	2890	2890	1926	1156	0%	60%	33%
Energy consumption	0.0999488	0.0999488	0.224872	0.134923	0%	55%	25%
Diagnosis Latency	5.71539	5.71539	17.1964	8.68424	0%	60%	33%

algorithm detects the faulty sensor nodes with more than 98% diagnosis accuracy for a wide range of fault probabilities and maintain a negligible (at max 6%) false alarm rate for lower connected network. However, the number of message exchanges, energy consumption, diagnosis latency and network life time remain same as that of the DSFDNC approach because both use neighbor coordination technique to diagnose the faulty sensor node. Therefore, in the next chapter, robust statistical test based approach is used to enhance the diagnosis accuracy and minimize the message complexity of the diagnosis algorithm

Chapter 5

Distributed Self Fault Diagnosis Algorithm for Large Scale WSNs *using* Modified Three Sigma Edit Test

Chapter 5

Distributed Self Fault Diagnosis Algorithm for Large Scale WSNs using Modified Three Sigma Edit Test

The classical methods for fault finding using mean, median, majority voting and hypothetical test based approaches are not suitable for large scale wireless sensor networks due to large deviation in inaccurate data transmission by different faulty sensor nodes. In this chapter, a modified three sigma edit test based self fault diagnosis algorithm is proposed which diagnose both hard and soft faulty sensor nodes. The proposed distributed self fault diagnosis algorithm is simulated in NS3 and the performances are compared with the existing distributed fault diagnosis algorithms. The simulation results show that the diagnosis accuracy, false alarm rate and false positive rate performance of the algorithm is better in adverse environment where the traditional methods fail to diagnose the fault. The other parameters such as diagnosis latency, energy consumption and the network lifetime are also determined.

5.1 Introduction

The fault diagnosis techniques based on classical estimates like sample mean, variance, co-variance or correlations are adversely influenced by large deviation of data for a faulty sensor node [6, 25, 56, 80]. These estimators are producing correct fault status when many sensor nodes are faulty within a particular region. Motivated by this, a modified three sigma edit test approach is adapted to diagnose the faulty sensor nodes present in wireless sensor networks (WSNs). In the proposed approach, the performance of the diagnosis depends on neighboring node's data where each sensor node participates in the fault diagnosis process to identify itself as faulty (hard or soft) or fault free. The accuracy in finding the fault status of all the sensor nodes depend on the number of neighboring nodes. We show that the DSFD3SET algorithm performs better when more number of neighboring nodes are likely to be faulty.

It has been seen from the literature and the algorithms discussed in previous chapters that the existing method leads to a large number of message exchanges over the network for data and fault status exchange. It puts a considerable overhead for the large scale WSNs. Due to poor performance and high energy overhead of the existing set of approaches, it is necessary to design and develop an efficient fault diagnosis algorithm for large scale WSNs.

This chapter has following contributions: (i) Modified three sigma edit test based fault diagnosis algorithm is discussed here to diagnose the faulty sensor nodes present in WSNs. (ii) A distributed self-fault diagnosis using Modified three sigma edit test (DSFD3SET) algorithm is developed where each sensor node diagnoses itself with high diagnosis accuracy and low false alarm rate and false positive rate. (iii) The proposed method is compared with traditional mean and three sigma edit test based fault diagnosis algorithm. (iv) Evaluation of the DSFD3SET algorithm using NS3 and comparing the performance with the existing works in the literature given by Chen *et al.* [6], Jiang [40] and the DSFDHT algorithm proposed in Chapter 4.

The remaining part of the chapter is organized as follows. The network, fault, and radio model are given in Section 5.2. The distributed self-fault diagnosis algorithm is presented in Section 5.3. The simulation results are given in Section 5.5. Finally, Section 5.6 concludes the chapter with discussions.

5.2 System Model

The system model discussed in Chapter 3 is similar to this chapter, except the fact that in this work, a highly dense WSN is considered where the average degree of the sensor nodes is high. In fact, a number of applications need to deploy a large number of sensor nodes in a small geographical area.

5.2.1 Assumptions, Notations, and Their Meanings

Similar assumptions are followed in this chapter as discussed in Chapter 3.

The list of notations and their meanings used in the DSFD3SET algorithm are given in Table 5.1.

Table 5.1: The notations used for developing the proposed DSFD3SET algorithm

Notation	Meaning
s_i	i^{th} sensor node
N	Total number of sensor nodes deployed
Neg_i	Set of neighboring nodes of s_i
A	Actual sensed data which remains same for all sensor nodes
$w_i(k)$	Erroneous data sensed by sensor node s_i
$x_i(k)$	Sensed data of s_i at time k
D	Maximum degree of WSNs
N_d	Diameter of WSNs
$\mathbf{Nx_i}$	Neighbors sensed data
NT_i	Neighboring table stored at s_i
FS_i	Fault status of s_i
θ	Threshold for identifying the sensor fault status
T_r	Transmission range of a sensor node s_i
$P_i(xco_i, yco_i)$	Position of s_i
N_i	Degree of s_i
$\hat{\mu}_i$	Sample mean of s_i
$\hat{\sigma}_i$	Sample variance of s_i
R	Breadth and width of the terrine of interest
ETT_i	Estimated transmission time of the sensor node s_i

5.2.2 Network and Radio Model

In this chapter, the network and radio model are same as specified in Chapter 3, except the fact that a dense sensor network is considered.

5.2.3 Fault Model

Let the WSN is considered as a random graph $G(S, C)$ where S represents set of independent and identically distributed sensor nodes and C represents the set of communication links between the sensor nodes. Let the set S_F represents the set of either hard or soft faulty sensor nodes. The fault free sensor nodes present in the network are $S_G = S - S_F$ where $N = |S_G + S_F|$ and $|S_F| << |S_G|$ respectively. Each sensor node is considered as a smart sensor having processing capability. Each sensor can diagnose itself as either faulty or fault free based on the fault status computed from it. For finding its fault status each sensor compares its battery power with a threshold (minimum battery power required for the sustainability of a sensor node which depends on the type of sensor circuits used in the sensor node). If it is less than the threshold, then the sensor node is declared as a hard faulty sensor node even if all the components of the sensor node are working properly. Apart from this, each sensor node can diagnose itself if its sensor circuit is not working properly by using the modified three sigma edit test based on the neighboring node's data.

The sensor nodes are assumed to be faulty when their actual sensed value deviates from their observed value (soft faulty sensor node) or do not respond to a request message (hard faulty sensor node). Every sensor node can be either subjected to a hard or soft fault but links are assumed to be fault free and taken care of by using error detecting and correcting codes which are usually implemented in the data link layer of the underlying networks.

The fault distribution is modeled assuming random distribution which is presented in Section 5.5. Each sensor node disseminates its own sensed data to its neighboring sensor nodes Neg_i and also collect the observation x_j from them. It requires storing the data in local memory for further use. The measured data may be temperature, humidity, wind speed, etc. sensed from the environment. Based on the normal and observed sensed values of different sensor nodes, the data is modeled as a normal distribution with a specific mean and standard deviation (SD). All the fault free sensor node's measured data is within acceptable range, whereas faulty node provides arbitrary values at different time.

5.3 Distributed Self Fault Diagnosis Algorithm using Modified Three Sigma Edit Test (DSFD3SET)

5.3.1 Description of the DSFD3SET Algorithm

The distributed self fault diagnosis (DSFD3SET) algorithm is consisting of two phases such as initialization and self-diagnosis phase. In the initialization phase, each sensor node s_i transmits a packet containing the sensed environmental data x_i to its neighboring nodes Neg_i and waits for an estimated transmission time ETT_i as derived in Equation (5.15). During that transmission time, it also collects all the packets coming from its neighboring nodes Neg_i. After ETT_i timeouts, each sensor node s_i extracts the information from all the receiving packets and maintains a neighboring Table NT_i. The table contains the detail information about the neighboring nodes id nid_j and their sensed information $\mathbf{Nx}_j$ which are given in Table 5.2. During this phase, all sensor nodes are assumed to be fault free and aware about their neighboring nodes Neg_i. In self-diagnosis phase, each sensor node s_i

Table 5.2: Neighboring table details

Si. No.	Node id(nid_j)	Sensed Data(Nx_{ij})	Final status(FS_{ij})
1	4	34.7	0
2	7	67.8	1
		.	
		.	
		.	
N	92	37.8	1

identifies hard faulty sensor nodes by comparing its own received data x_i with the neighboring table (NT_i)'s data. When sensor node s_i does not receive any data for a neighboring node s_j, it assumes that sensor node s_j is hard faulty. If the sensor node s_i receives data from s_j then it performs the modified three sigma edit test (as discussed in Section 5.3) over the received data $\mathbf{Nx_i}$ and its sensed data x_i to identify the soft faulty neighboring sensor nodes and its fault status. If a sensor node s_i is neither diagnosed as hard nor soft faulty, then it is assumed as fault free. The detail description about this algorithm is summarized in Algorithm 5.1.

Algorithm 5.1 DSFD3SET Algorithm

Data: N_i Nodes, $\mathbf{Nx}_i$
Result: Calculate S_F and S_G
Initialize $S_F = \phi$, and $S_G = \phi$
Each sensor node s_i collects environmental sensed data $\mathbf{Nx_i}$ from the neighbors Neg_i and construct a neighboring table NT_i.
Set $FS_i = 0$(Fault free).
for $j = 1 \cdots |Neg_i|$ **do**
  $FSNeg_{i,j} = 0$
end
Step 1. After ETT_i time expires
s_i identifies list of hard fault sensor nodes M (say) by comparing NT_i and data collected from Neg_i.
Step 2. Calculation of self fault status along with the status of neighboring nodes.
$\mathbf{Nx_i} = \mathbf{Nx_i} \cup \{x_i\}$
Sort($\mathbf{Nx_i}$)
/* Procedure for sorting all the elements of $\mathbf{Nx_i}$ in ascending order*/
if $|Neg_i|\%2 == 0$ **then**
  $md_i = [\mathbf{Nx_i}[|Neg_i|/2] + \mathbf{Nx_i}[(|Neg_i| + 1)/2]]/2$
else
  $md_i = [\mathbf{Nx_i}[|Neg_i|/2]$
end
$ADM = \phi$
for $j = 1 \cdots |Neg_i| + 1$ **do**
  $ADM_i = ADM_i \cup (\mathbf{Nx_i}[j] - md_i)$
end
if $|ADM_i|\%2 == 0$ **then**
  $mad_i = [ADM_i[|Neg_i|/2] + ADM_i[(|Neg_i| + 1)/2]]/2$
else
  $mad_i = [ADM_i[|Neg_i|/2]$
end
$MADN_i = mad_i/0.675$
$FSC_i = (x_i - md_i)/MADN_i$
if $FSC_i < 3$ **then**
  $FS_i = 0 \quad S_G = S_G \cup \{s_i\}$
else
  $FS_i = 1 \quad S_F = S_F \cup \{s_i\}$
end
for $j = 1 \cdots |Neg_i|$ **do**
  $FSC_i = (\mathbf{Nx_i}[j] - md_i)/MADN_i$
  if $FSC_i < 3$ **then**
    $FSNeg_{i,j} = 0 \quad S_G = S_G \cup \{s_j\}$
  else
    $FSNeg_{i,j} = 1 \quad S_F = S_F \cup \{s_j\}$
  end
end

5.4 Analysis of the DSFD3SET Algorithm

The proposed DSFD3SET algorithm for WSNs is based on the assumption of the network and fault model given in the previous Section 5.2. Let the data of sensor node s_i at k^{th} time instant is denoted as $x_i(k)$. In order to find the faulty sensor nodes in the network, we need to analyze the data $\{x_i(k)\}_{i=1}^{N}$ of all the neighboring sensor nodes. The sensor reading $x_i(k)$ can be either actual sensed data or the erroneous data. This $x_i(k)$ follows normal distribution $\mathcal{N}(0, \sigma_i^2)$ where σ_i^2 is the variance of erroneous data present at sensor node s_i. The analytical model for sensor node's data is written in Equation (5.1) as

$$x_i(k) = A + w_i(k) \quad \text{where} \quad i = 1, 2, 3, \cdots, N \tag{5.1}$$

where A be the actual data (like temperature, pressure, humidity, etc.) measured by the sensor node s_i and $w_i(k)$ is the erroneous data due to environmental noise or distortion in signal [95]. Here the erroneous data are assumed to be temporally and spatially independent. The model assumes that all the sensor nodes measured same actual data, but the magnitude of erroneous data of different sensors are different [100]. The *probability density function*(pdf) of x_i is given by

$$f_X(x_i(k)) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\frac{-(x_i(k)-A)^2}{2\sigma_i^2}} \tag{5.2}$$

In general, for homogeneous network, the data variance is assumed to be same for all the fault free sensor nodes and denoted as σ^2. But, the variance of the faulty sensor nodes measurement data is very high (about 100 times the variance of fault free sensor node) and denoted as σ_f^2.

Let N_i represent the degree of sensor node s_i which is defined as number of neighboring sensor nodes of s_i and is given in Equation (5.3).

$$N_i = \sum_{\forall s_j \in S, i \neq j} dist(s_i, s_j) \leq T_r, \quad j = 1, 2, 3, 4, \ldots, N \tag{5.3}$$

where $s_i, s_j \in S$, $dist(s_i, s_j)$ is the Euclidean distance between s_i and s_j, and T_r is the transmission range of every sensor node s_i.

In distributed self fault diagnosis approach, each sensor node s_i accumulates the data set $\mathbf{Nx}_i$ from one hop neighbors s_i and s_j and represents as $\mathbf{Nx}_i = \{x_j\}_{s_j \in Neg_i}$. In order to find the fault status, the mean and variance of the data set $\mathbf{Nx}_i$ is determined. The sample mean $\hat{\mu}_i$ and standard deviation $\hat{\sigma}_i$ at sensor node s_i are defined in Equation (5.4) and Equation (5.5) respectively.

$$\hat{\mu}_i = \frac{1}{N_i} \sum_{s_j \in Neg_i} x_j \tag{5.4}$$

and

$$\hat{\sigma}_i = \sqrt{\frac{1}{N_i - 1} \sum_{s_j \in Neg_i} (x_j - \hat{\mu}_i)^2} \tag{5.5}$$

Each sensor node s_i can analyze its fault status after estimating the mean $\hat{\mu}_i$ and standard deviation (SD) from Equation (5.4) and Equation (5.5) respectively.

In this method, initially the presence of faulty sensor node in the neighbor is determined by observing the estimated mean or standard deviation. Basically the mean and SD provides the location and deviation of the data. These classical estimates are influenced by the presence of erroneous data provided by a faulty sensor node. A single erroneous data has an unbounded influence (both the mean and SD varying form $-\infty$ to ∞) on these two classical estimation. If the measured mean and SD is beyond the confidence interval, then there is a presence of outlier (i.e. the presence of the faulty sensor node provides the outlier data) [101].

In order to find the status of the sensor nodes, the traditional algorithms use the comparison model [6, 40] in which each sensor node compares own data with their neighbor's data. The demerits of using this method is that it finds two faulty sensor nodes as fault free, if the variation of the data between two faulty sensor nodes is less. Both of them are erroneously diagnosed as fault free. In order to avoid this possibility, we propose a statistical measure of the outlyingness of an observation of a sensor node x_i with respect to an estimated mean $\hat{\mu}_i$.

$$d_i = |x_i - \hat{\mu}_i| \tag{5.6}$$

Where d_i is the dispersion between x_i and $\hat{\mu}_i$ of the sensor node s_i. The sensor node itself is identified as faulty if $d_i > \theta$ (threshold) otherwise it is fault free. It is assumed that the observed data values follow a normal distribution, the threshold θ can be defined in terms of variation in the erroneous data. For example, if $\theta = 3\sigma$ where σ is the standard deviation of the erroneous data, then there is a 99.73% probability that the observation lies in between $\hat{\mu}_i - 3\sigma$ to $\hat{\mu}_i + 3\sigma$. The mean used here deviates from the true value if one sensor node's data deviate from the actual data beyond a range.

The outlyingness of an observation can be measured by taking both estimated mean and variance of the data collected from the neighbors. The outlyingness t_i is the ratio between its deviation to the estimated mean $\hat{\mu}_i$ and SD. This is calculated as

$$t_i = \frac{x_i - \hat{\mu}_i}{\hat{\sigma}_i} \tag{5.7}$$

According to the 'three-sigma-edit' rule, if $|t_i| > 3$ for any sensor node s_i it is deemed as suspicious and regarded as a faulty sensor node. Otherwise the sensor node s_i is computed as fault free. A single faulty node has a serious adverse influence on any kind of WSNs applications like parameter estimation or event detection. It also influences the mean and SD estimation. A more accurate fault diagnosis algorithm design aims at minimizing the performance degradation due to model errors or uncertainties. Certainly, the robust performance is an accurate parameter estimation which exploits all available information about the sensor network. But, we need to find algorithms which are robust, i.e., less sensitive to the remaining model uncertainties.

Further, this traditional rule has some drawbacks. The rule is ineffective for a small number of samples. If $N < 10$, then $|t_i|$ is always less than 3. That shows that for lower degree network, this rule is unable to notice the faulty sensor node. When there are several faulty sensor nodes, their effects may interact in such a way that some faulty sensor nodes remain unnoticed as faulty. For example, if two sensor nodes data are suspicious and one sensor node data is very large compared to that of another sensor node. In such case, the faulty sensor node having comparable lower value than the other faulty sensor node is noticed as good sensor node. This effect is called *masking*.

One can measure the median of the data by finding the median instead of the mean. The median of the data set $\mathbf{x}_k = \{x_1, x_2, ..., x_{n_k}\}$ is calculated after sorting the observation in increasing order as

$$x_{(1)} \leq x_{(2)} ... \leq x_{(n_i)}$$

If n_i is odd, then $n_i = 2m-1$ for some integer m and the median $Md_i = \text{Med}(\mathbf{x}_i) = x_{(m)}$. Similarly if n_k is even and is given as $n_k = 2m$ for some integer m, the median is defined as

$$Md_i = Med(\mathbf{x}_i) = \frac{x_{(m)} + x_{(m+1)}}{2} \tag{5.8}$$

In the literature, many authors have used median based fault finding techniques [26]. Though median based computation is complex due to need of sorting, it performs better accuracy in results. This is due to the deviation of actual data value from the faulty sensor reading. Similarly, another alternative to the SD is the median of the absolute deviation (around the mean) [101] of the observation from the median and it is known as median absolute deviation (MAD) [102]. This is defined as

$$MAD(x_1, ..., x_{n_i}) = \text{Med}|\mathbf{x}_i - Md_i| \tag{5.9}$$

In order to use the MAD as like SD, the normalized median absolute deviation (about the median) MADN($\mathbf{x}_i$) is used which is defined as

$$\text{MADN}(\mathbf{x}_i) = \frac{\text{Med}\{|\mathbf{x}_i - Md_i)|\}}{0.675} \tag{5.10}$$

To avoid the drawback of the discussed fault diagnosis method based on $\hat{\mu}$ and SD, the mean $\hat{\mu}_i$ is replaced by the median of the neighbor's data Med($\mathbf{x}_i$). In the place of SD, we consider the normalized median absolute deviation about the median (MADN($\mathbf{x}$)). Now, one new measure of outlyingness t_i^r is defined as

$$t_i^r = \frac{x_i - Md_i}{\text{MADN}(\mathbf{x}_i)} \tag{5.11}$$

where t_i^r is the absolute error for the modified three sigma edit test.

The t_i^r defined in Equation (5.11) is known as 'modified three-sigma edit rule'. This method is accurate to diagnose the faulty sensor nodes when the number of faulty sensor nodes in WSN is more. Therefore, the DSFD3SET algorithm is suitable for a large number of faulty sensor nodes present in highly dense WSN.

The parameters such as distance, estimated transmission time, diagnosis accuracy, false alarm rate, false positive rate, diagnosis latency, message complexity, energy complexity, and network life time are considered to evaluate the performance of the proposed DSFD3SET algorithm. The following lemmas i.e., Lemma 5.1 through Lemma 5.9 along with their proofs are presented below for analytical evaluation of the proposed DSFD3SET algorithm.

Lemma 5.1: Euclidean distance $dist(s_i, s_j)$ between any pair of sensor nodes s_i and s_j in WSNs is $\frac{k}{\sqrt{P_r}}$ where P_r is the received power and k is a constant (depends upon the parameters set up by the transceiver system).

Proof

To calculate the Euclidean distance $dist(s_i, s_j)$ between any pair of sensor nodes s_i and s_j, we need the physical location of each pair of sensor nodes s_i and s_j. If all the sensor nodes broadcast their physical location, then the energy of the sensor node depletes and traffic of the network increases. To avoid such situation, we calculate the approximate Euclidean distance between any two sensor nodes s_i and s_j by considering the Friss propagation loss model [103].

In Friss free space propagation loss model, the received power P_r is computed as

$$P_r = P_t \times G_t \times G_r \times \frac{\lambda^2}{(4 \times \pi \times D)^2} \tag{5.12}$$

where P_r is the power received by the receiving antenna, P_t is the power transmitted by the transmitting antenna, G_t and G_r are the gain of transmitting and receiving antenna, λ is the wavelength of the signal and D is the distance between transmitting and receiving antenna. An approximate distance D between any two sensor nodes s_i and s_j are calculated by using Equation (5.12) as

$$dist(s_i, s_j) = \frac{k}{\sqrt{P_r}} \tag{5.13}$$

where k is a constant which is given as

$$k = \sqrt{P_t \times G_t \times G_r \times \frac{\lambda^2}{(4 \times \pi)^2}} \tag{5.14}$$

This proves Lemma 5.1.

Lemma 5.2: The estimated transmission time (ETT_i) for sensor node s_i in WSNs is $\frac{dist(s_i,s_j)}{l} + \tau_i$ where, τ_i is the processing time and l is speed of light.

Proof

Estimated transmission time (ETT_i) is the approximate time required by a sensor node s_i to transmit its data to all its surrounding neighbors nodes Neg_i which are coming under its transmission range T_r. The estimated transmission time ETT_i is defined as

$$ETT_i = \{\max\{ETT_{i,j} + \tau_i, \forall s_j \in Neg_i\}\} \tag{5.15}$$

where $ETT_{i,j}$ i.e. estimated transmission time between the sensors s_i and s_j and τ_i is the processing delay of s_i. The $ETT_{i,j}$ is calculated as given in Equation (5.16).

$$ETT_{i,j} = \frac{dist(s_i, s_j)}{l} \tag{5.16}$$

where l is the speed of light. This proves Lemma 5.2.

Lemmas 5.1 and 5.2 are used for diagnosing the hard faulty sensor nodes and modified three sigma edit test is used for diagnosing the soft faulty sensor nodes present in the network. Lemmas 5.3 to 5.5 are used for estimating the performance of the DSFD3SET algorithm.

Lemma 5.3 The diagnosis accuracy (DA) of the DSFD3SET algorithm is

$$DA = \begin{cases} p_r - p, \ t_i^r \leq \theta \\ (1 - p_r) + p, \ t_i^r > \theta \end{cases} \tag{5.17}$$

Proof: The diagnosis accuracy is defined as the ratio between the number of faulty sensor nodes diagnosed as faulty and the total number of faulty sensor nodes present in the network. Suppose p_r is the probability that a sensor node is faulty, $1-p_r$ is the probability that a sensor node is fault free. p is the probability that a faulty sensor node detected as fault free. Therefore, the diagnosis accuracy is $p_r - p$ when $t_i^r \leq \theta$ where t_i^r is absolute error for modified three sigma edit test and θ is the threshold value based on the accuracy. When $t_i^r > \theta$, the diagnosis accuracy is $(1 - p_r) + p$. This proves Lemma 5.3.

Lemma 5.4: The false alarm rate (FAR) of DSFD3SET algorithm is

$$FAR = \begin{cases} \frac{p_r}{1-p_r},\ t_i^r \leq \theta \\ \frac{q}{1-p_r},\ t_i^r > \theta \end{cases} \tag{5.18}$$

Proof: The false alarm rate is defined as the ratio of the number of fault free sensor nodes diagnosed as faulty to the total number of fault free sensor nodes present in the network. Suppose p_r is the probability that a sensor node is faulty, $1-p_r$ is the probability that a sensor node is fault free. q is the probability that a faulty sensor node diagnosed as fault free. Therefore, the false alarm rate is $\frac{p_r}{1-p_r}$ when $t_i^p \leq \theta$ where t_i^r is absolute error for modified three sigma edit test and θ is the threshold value based on the accuracy. Otherwise, when $t_i^r > \theta$, the false alarm rate is $\frac{q}{1-p_r}$. This proves Lemma 5.4.

Lemma 5.5 : The False positive rate (FPR) for the DSFD3SET algorithm is

$$FPR = \begin{cases} p_r + p,\ t_i^r \leq \theta \\ 1-(p_r+p),\ t_i^r > \theta \end{cases} \tag{5.19}$$

Proof: The false positive rate is defined as the ratio of the number of faulty sensor nodes diagnosed as fault free to the total number of faulty sensor nodes present in the network. Suppose p_r is the probability that a sensor node is faulty, $1-p_r$ is the probability that a sensor node is fault free. p is the probability that a faulty sensor node detected as fault free. Therefore, the false positive rate is $p_r + p$ when $t_i^p \leq \theta$ where t_i^r is absolute error for modified three sigma edit test and θ is the threshold value based on the accuracy. Otherwise, when $t_i^p > \theta$, the FPR is $1-(p_r+p)$. This proves Lemma 5.5.

Lemma 5.6: The diagnosis latency (DL) of DSFD3SET is $O(ETT + P_T)$ where ETT is the maximum expected transmission time of the network and P_T is the processing time.

Proof

The DL of the DSFD3SET algorithm is defined as the total amount of time required to diagnose all the sensor nodes present in the network. In the DSFD3SET algorithm, after broadcasting their own sensed data, each sensor node s_i waits a fixed expected transmission time ETT_i. After that time expires each sensor node s_i starts their processing task of identifying the hard and soft faulty sensor nodes present around it.

Since the sensor nodes are homogeneous in nature (as discussed in Section 5.2), for processing, it needs constant time P_T (say). Therefore, total time consumed by the sensor node s_i is $P_T + ETT_i$. The ETT_i is different for each sensor node and all the sensor nodes execute the task simultaneously. It needs maximum $P_T + ETT$ time, where ETT is the $\max\{\forall\{ETT_i\}\}$ where $i = 1, 2, 3, \ldots N$. This proves Lemma 5.6.

Lemma 5.7: The message complexity (MC) of the DSFD3SET algorithm is $O(N)$ where N is the number of sensor nodes in WSNs.

Proof

The MC is nothing but the total number of messages exchanged by the sensor nodes over the network for executing the DSFD3SET algorithm. In self diagnosis phase, the DSFD3SET algorithm needs one message exchange over the network, which means each individual sensor node s_i broadcast a single message (i.e. the own sensed data) to their neighbors Neg_i. Based on the received data, each sensor node s_i calculates its own fault status by applying modified three sigma edit test method. Therefore, the message complexity of DSFD3SET algorithm is $O(N)$. This proves Lemma 5.7.

Lemma 5.8: The total amount of energy consumption (TEC) by the DSFD3SET algorithm ($EC_{DSFD3SET}$) is $\sum_{s_i \in S}(E_T(m.d) + N_i \times E_R(m, d))$.

Proof

In the DSFD3SET algorithm, each sensor node s_i broadcasts its own sensed data x_i over the network for which $E_T(m, d)$ units of energy are required. Each sensor node s_i receives that data from their neighbors for which it requires $N_i E_R(m, d)$ units of energy. Therefore, the total amount of energy required by the sensor node s_i is $E_T(m, d) + N_i \times E_R(m, d)$. The TEC of WSN is

$$TEC = \sum_{s_i \in S}(E_T(m, d) + N_i \times E_R(m, d)) \tag{5.20}$$

where $E_T(m, d) = m \times (\alpha_1 + \alpha_2 \times d^{\alpha})$ and $E_R(m, d) = m \times \alpha_3$ (as discussed in radio model of Section 5.2). This proves Lemma 5.8.

Lemma 5.9: The network life time (NLT) for the DSFD3SET algorithm is $min\{TE/CE_i\}$ $i = 1, 2, 3, \ldots N$ where TE and CE_i are the total energy assigned to every sensor node s_i and the total energy consumed by the sensor node s_i.

Proof:

The NLT is the time required for the total number of data gathering rounds which cause the first sensor node of the network to die due to energy consumption. As all the sensor nodes are uniform in nature (as discussed in Section 5.2), the sensor node s_i utilizes PE_i amount of energy for data processing, $EC_{DSFD3SET_i}$ amount of energy for fault diagnosis and $EOAN_i$ amount of energy for normal activity of the network. Therefore, the network life time NLT of the DSFD3SET algorithm is $min\{\frac{TE}{CE_i}\}$ where $CE_i = PE_i + EC_{DSFD3SET_i} + EOAN_i$. This proves Lemma 5.9.

5.4.1 An Example

In this section, the working mechanism of the DSFD3SET algorithm is illustrated through an example. The aim is to show the accuracy of the modified three sigma edit test method over traditional methods such as mean and three sigma edit test methods. Let us consider a k-connected sensor network of having maximum degree 10 i.e., there are 10 neighboring nodes for a sensor node ($N_k = 10$). Each sensor node measures the environmental temperature and then share with the immediate neighbors. The sensor node's data are generated by using the model given in Equation (5.1). Consider high erroneous environment and the data variance in each of the fault free sensor nodes is $\sigma_g^2 = 1$ and high variance i.e, $\sigma_f^2 = 1000$ is for a faulty sensor node. Initially, all the sensor nodes are assumed to be fault free and the data (temperature in degree centigrade) are given in Table 5.4. Four sensor nodes with node id 2, 5, 7, and 9 are used as faulty neighboring nodes. The statistical parameters are measured by using the formula given above are provided in Table 5.3.

The fault status of each of the sensor nodes is calculated by following three different methods described in Section 5.3. In Method 1, the absolute difference between mean and own measured data is used for finding the fault. Method 2

Table 5.3: Statistical parameters of 10 sensor nodes with and without fault

Parameter	Without fault	With fault
Mean ($\hat{\mu}$)	25.653	31.808
Median (Md)	25.716	26.453
Standard Deviation ($\hat{\sigma}$)	0.846	175.705
Median Absolute Deviation (MAD)	0.737	2.119
Normalized Absolute Deviation ($MADN$)	1.0925	3.1396

involves the 3-sigma rule for detecting the faulty sensor node. The modified three sigma edit test is used in Method 3. The data of sensor nodes after occurrences of faults, the outlying measures in three different methods and the estimated fault status are provided in Table 5.4. In all cases, the outlyingness is compared with 3σ.

Table 5.4: Estimated fault status of 10 sensor nodes by Methods 1,2 and 3

Node No.	IFS	data	$\|x-\hat{\mu}\|$	FSD1	$\|t_k(5.7)\|$	FSD2	$t_k^r(5.11)$	FSD3
1	0	25.715	6.093	0	0.459	0	0.235	0
2	1	40.255	8.446	0	0.637	0	4.39	1
3	0	24.876	6.932	0	0.523	0	0.502	0
4	0	26.489	5.318	0	0.401	0	0.011	0
5	1	59.417	27.320	0	2.060	0	10.407	1
6	0	26.417	5.392	0	0.406	0	0.011	0
7	1	48.658	16.849	0	1.271	0	7.07	1
8	0	23.792	8.0158	0	0.604	0	0.849	0
9	1	16.122	15.686	0	1.183	0	3.29	1
10	0	26.630	5.178	0	0.390	0	0.056	0

0: Status for fault free sensor node
1: Status for faulty sensor node
FSD1, FSD2 and FSD3: Fault status detected by using methods 1,2 and 3 respectively

It is clearly shown in Table 5.4 that both the Methods 1 and 2 fails to detect at least one faulty sensor node out of four faulty sensor nodes present in the neighbors. It is because these methods used mean and standard deviation to measure the outlyingness which is not accurate due to faulty sensor nodes. Whereas in method 3 the modified three sigma edit test, the statistical parameters (M_d and $MADN$) are used which are more accurate to the erroneous data. Thus, the method 3 is detecting all the faulty sensor nodes as faulty. The deviation of the statistical parameters when fault is occurring are given in Table 5.3. When one faulty sensor node's data is too high compared to other, then the 3-sigma rule can detect only the faulty sensor node providing very high value and cannot detect the second one. Therefore, the large value acts as a mask to hide the small value.

5.5 Simulation Results and Discussions

The performance of the DSFD3SET algorithm is measured by calculating the generic performance parameters such as diagnosis accuracy, false alarm rate, false positive rate, energy consumption, diagnosis latency, and network life time which are defined in Section 5.3. The DSFD3SET algorithm is simulated in NS3 [38] and the performances are compared with existing algorithms such as DFD, IDFD, and DSFDHT algorithms using the above parameters. The simulation parameters used in NS3 simulator are provided in Table 5.5. The algorithms are tested for different fault

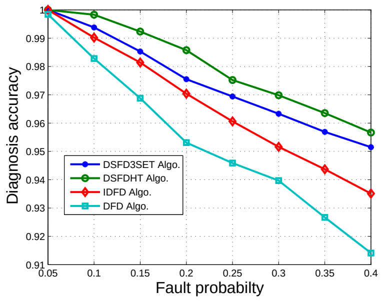
Table 5.5: Simulation parameters

Parameter	Value
Network size	512 sensor nodes
Average degree	10, 15, 20, 25
Topologies	Arbitrary network
Propagation Loss Model	Range propagation loss model
MAC	IEEE 802.15.4
Simulation time	300s
α_1	50 nJ/bit
α_2	10 pJ/bit/m^2
α_3	50 nJ/bit
T_r	$(56, 61, 68, 74)m$
Network Grid	From $(0, 0)$ to $(500, 500)m$
Initial Energy	1J

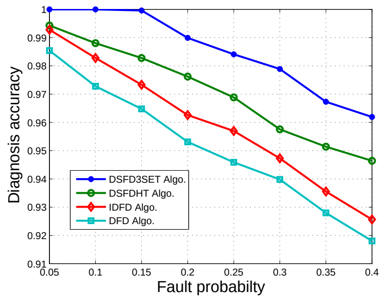
probabilities from 0.05 to 0.4 in the step size of 0.05. Since the modified three sigma edit test method's performance depends on the number of data, therefore the algorithm is verified for different average degrees. In order to get the average of the degree of all sensor nodes in WSNs from 10 to 25 with step size of 5, the transmission ranges are chosen $40, 54, 60$, and 67 respectively. We have performed 100 experiments for each point of the plot and average is plotted. The performance of the DSFD3SET algorithm is compared with the DSFDHT algorithm [Chapter 4] and the existing algorithms [6] (DFD Algorithm) and [40] (IDFD Algorithm).

5.5.1 The diagnosis accuracy, false positive rate and false alarm rate Performance

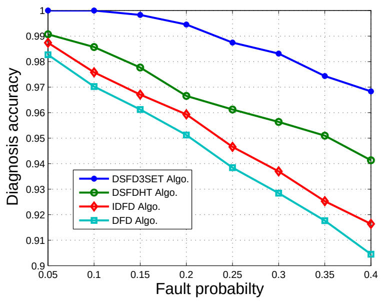
The diagnosis accuracy, false positive rate and false alarm rate performances with respect to fault probabilities for different average degrees of the network are plotted in Figure 5.1, Figure 5.2 and Figure 5.3 respectively. As we see from the figures that



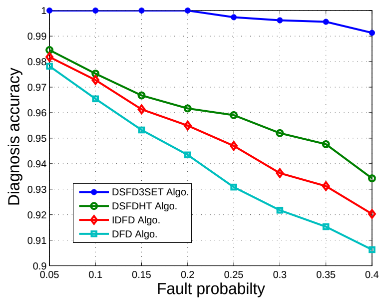
(a) For average degree $N_a = 10$



(b) For average degree $N_a = 15$



(c) For average degree $N_a = 20$

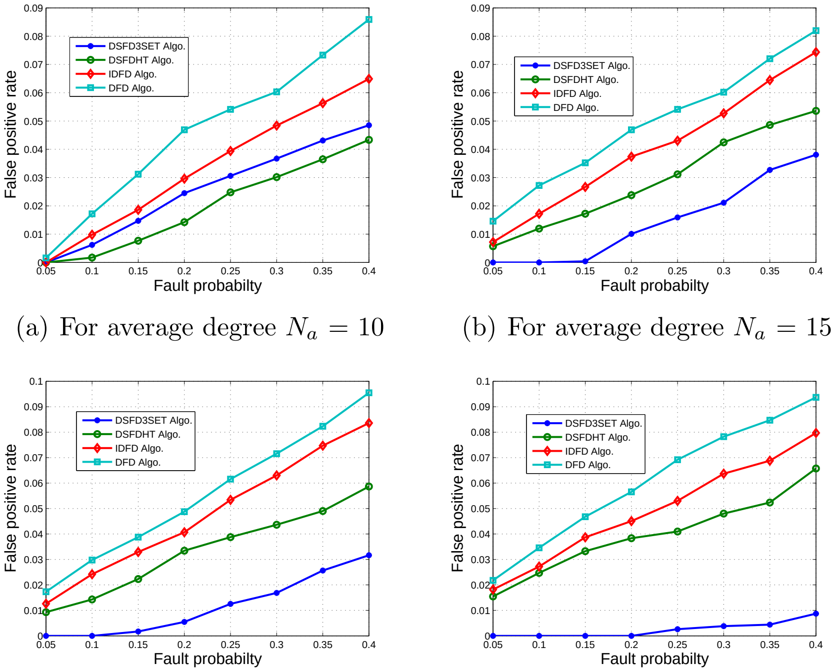


(d) For average degree $N_a = 25$

Figure 5.1: Diagnosis accuracy versus fault probability plots for the DSFD3SET, DSFDHT, DFD and IDFD algorithms.

the DSFD3SET algorithm yields significant performance over other algorithms with 100% diagnosis accuracy and 0% false alarm rate and false positive rate when N_a is more than 15 with fault probability less than 20%. The diagnosis accuracy, false positive rate and false alarm rate of the DSFD3SET algorithm are nearly 100% and zero for high fault probability (up to 30%) and higher (More than $N_a = 25$) average degree. Whereas, the DSFDHT, DFD, and IDFD algorithms have less performance .

The diagnosis accuracy decreases and false alarm rate, and false positive rate increase when fault probabilities increase as shown in Figure 5.1, Figure 5.2 and Figure 5.3 respectively. In the worst case scenario, (when 40% sensor nodes are faulty and $N_a = 10$), the DSFD3SET algorithm is able to diagnose 95% of faulty sensor nodes unlike the DSFDHT, DFD and IDFD algorithms which have diagnosis accuracy of 96%, 94%, and 92% respectively. The false positive rate is 5% for DSFD3SET algorithm whereas the DSFDHT, DFD, and IDFD algorithms which have 4%, 7%, and 9% respectively. Similarly, the false alarm rate is 3% for DSFD3SET algorithm and



(a) For average degree $N_a = 10$ (b) For average degree $N_a = 15$

(c) For average degree $N_a = 20$ (d) For average degree $N_a = 25$

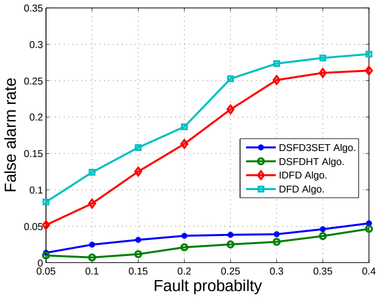
Figure 5.2: False positive rate versus fault probability plots for the DSFD3SET, DSFDHT, DFD and IDFD algorithms.

the DSFDHT, DFD and IDFD algorithm give 2%, 5%, and 9% respectively. The performance increases when the average degree increases for all the algorithms, including DSFD3SET, however the number of message transmissions remain constant as one sensor node transmits only one message for diagnosis purpose.

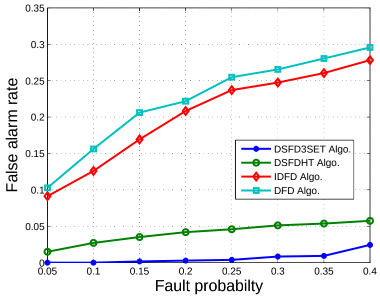
The accurate fault diagnosis is observed in DSFD3SET algorithm because, the outlyingness of the faulty sensor node's data is measured by modified three sigma edit test based method and the parameters used to estimate the outlyingness are accurate to the presence of erroneous data produced by faulty sensor nodes. Whereas other comparison based method, the parameters used for comparison are deviating when a faulty sensor node is present in the neighborhood.

5.5.2 Diagnosis accuracy and false alarm rate Analysis with Respect to Confidence Interval

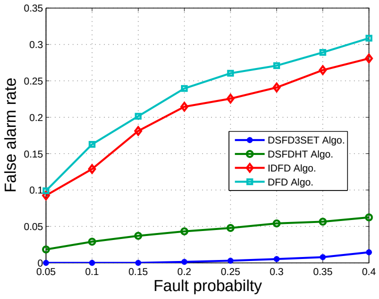
The 95% confidence interval (CI) of diagnosis accuracy and false alarm rate for different fault probabilities (p_r), and average degrees are provided in Table 5.6 and Table 5.7 respectively. From the tables, it is shown that the CI is less for the



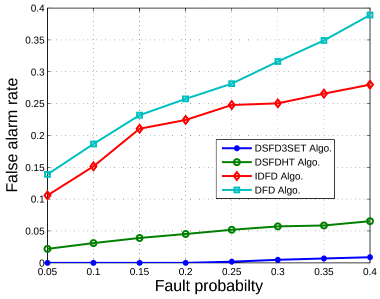
(a) For average degree $N_a = 10$



(b) For average degree $N_a = 15$



(c) For average degree $N_a = 20$



(d) For average degree $N_a = 25$

Figure 5.3: False alarm rate versus fault probability plots for the DSFD3SET, DSFDHT, DFD and IDFD algorithms.

DSFD3SET algorithm as compared to that of other algorithms. The CI increases when the fault probability of the network increases for all the algorithms. However, the CI decreases when the average degree of the network N_a increases. For more number of data, the fault decision will be more accurate. The CI performance for the DSFDHT, DFD and IDFD algorithms is not improved when average degrees are substantially increased.

The range of values between the minimum and maximum of false alarm rate in percentage when $p_r = 0.3$, and $N_a = 25$ is 0.4561 to 1.5734, 2.2874 to 9.0652, 19.3078 to 31.0459 and 25.5010 to 38.1862 with respect to CI of 95% for the algorithm DSFD3SET, DSFDHT, IDFD and DFD respectively. Similarly, The range of values between the minimum and maximum of diagnosis accuracy in percentage when $p_r = 0.3$, and $N_a = 25$ is 100.0000 to 100.0000, 90.4759 to 99.5241, 89.0702 to 98.9298, and 86.3685 to 97.6315 with respect to 95% CI for the algorithm DSFD3SET, DSFDHT, IDFD, and DFD respectively. This clearly shows that the proposed algorithm outperforms over the existing algorithms. Even though, when $p_r = 0.3$ and $N_a = 25$, the diagnosis accuracy is 100%, however the existing algo-

Table 5.6: Confidence interval of diagnosis accuracy for DSFD3SET, DSFDHT, IDFD and DFD algorithms

Fault Probability	Algorithm	$N_a = 15$	$N_a = 20$	$N_a = 25$
p_r=0.2	DSFD3SET	(96.4745,100.00)	(100.000,100.0000)	(100.0000,100.0000)
	DSFDHT	(94.4465,100.00)	(92.6701,100.0000)	(92.6701,100.000)
	IDFD	(92.6701,100.00)	(91.0261,100.000)	(91.0261,100.00)
	DFD	(91.0261,100.00)	(89.4681, 100.000)	(89.4681,100.000)
p_r=0.25	DSFD3SET	(94.8124,100.000)	(96.7345,100.0000)	(100.0000,100.0000)
	DSFDHT	(94.8124,100.000)	(93.1159,100.0000)	(91.5382,100.000)
	IDFD	(91.5382,100.000)	(90.0376,99.9624)	(90.0376,99.9624)
	DFD	(90.0376,99.9624)	(90.0376,99.9624)	(88.5927,99.4073)
p_r=0.3	DSFD3SET	(95.0939,100.00)	(96.9346, 100.0000)	(100.0000,100.0000)
	DSFDHT	(91.9323,100.00)	(91.9323,100.0000)	(90.4759,99.5241)
	IDFD	(90.4759,99.5241)	(89.0702,98.9298)	(89.0702,98.9298)
	DFD	(89.0702,98.9298)	(87.7037,98.2963)	(86.3685,97.6315)

Table 5.7: Confidence interval of false alarm rate for DSFD3SET, DSFDHT, IDFD and DFD algorithms

Fault Probability	Algorithm	$N_a = 15$	$N_a = 20$	$N_a = 25$
p_r=0.2	DSFD3SET	(0.3995,1.3775)	(0.3846,0.8736)	(0,0)
	DSFDHT	(1.7883,7.0137)	(1.7883,7.0137)	(1.5146,7.3263)
	IDFD	(15.8364,26.2174)	(16.2816,26.7501)	(16.1057,27.8123)
	DFD	(16.9516,27.5472)	(18.5239,29.3978)	(20.3356,31.4982)
p_r=0.25	DSFD3SET	(0.4254,1.4671)	(0.4254,1.4671)	(0.4095,0.9304)
	DSFDHT	(1.9089,7.4661)	(2.0971,7.7988)	(1.9646,8.1292)
	IDFD	(18.1080,29.2878)	(17.1534,28.1591)	(17.1754,30.6922)
	DFD	(19.7896,31.2520)	(20.5144,32.0897)	(22.2146,34.0354
p_r=0.3	DSFD3SET	(0.4031,2.0791)	(0.4561,1.5734)	(0.4561,1.5734)
	DSFDHT	(2.2552,8.3594)	(2.4598,8.7134)	(2.2874,9.0652)
	IDFD	(18.9761,30.7446)	(18.4623,30.1410)	(19.3078,31.0459)
	DFD	(20.7844,32.8469)	(21.0440,33.1460)	(25.5010,38.1862)

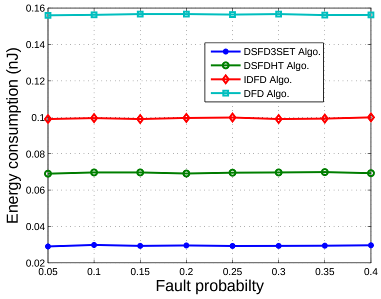
rithm's performance is not satisfactory. Therefore, from the tables it has been seen that the CI performance of the DSFD3SET is better as compared to that of the DSFDHT, IDFD, and DFD algorithms.

Table 5.8: Total number of messages exchanged for DSFD3SET, DSFDHT, DFD, IDFD algorithms

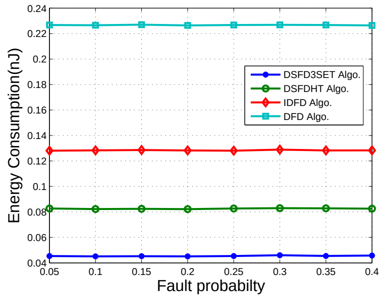
Average degree (N_a)	DSFD3SET Algorithm	DSFDHT Algorithm	DFD Algorithm	IDFD Algorithm
N_a = 10	512	1024	2560	1536
N_a = 15	512	1024	2560	1536
N_a = 20	512	1024	2560	1536
N_a = 25	512	1024	2560	1536

5.5.3 Message Complexity

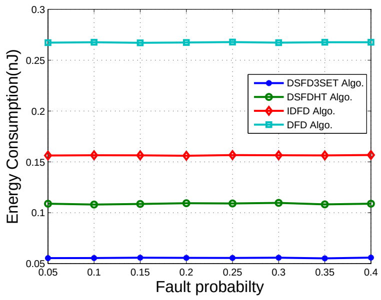
The fault diagnosis algorithm DSFD3SET has less message overhead as compared to that of existing algorithms. Total number of messages exchanged depend on the number of sensor nodes present in the network, the degree of the sensor nodes and number of times message exchange required to find the fault status. The message complexity is independent of fault probability, because in fault diagnosis method, it is assumed that all the sensor nodes send a request message and expect a response message with respect to their neighboring nodes by using one hop communication. The algorithms DSFDHT, DFD and IDFD incur more messages overhead as com-



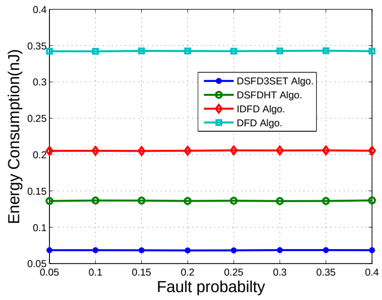
(a) For average degree $N_a = 10$



(b) For average degree $N_a = 15$



(c) For average degree $N_a = 20$



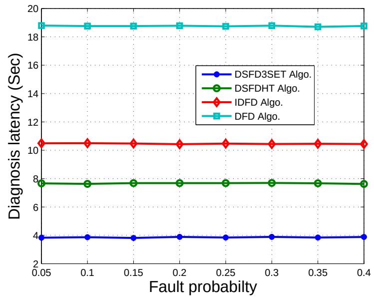
(d) For average degree $N_a = 25$

Figure 5.4: EC versus fault probability plots for the DSFD3SET, DSFDHT, DFD and IDFD algorithms.

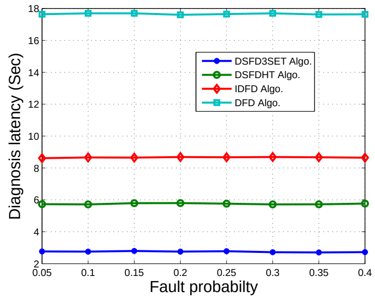
pared to DSFD3SET. It is because, the DSFD3SET algorithm needs only 1 message for identifying the fault status of the sensor node. The DSFD3SET algorithm is accurate and the sensor nodes are able to diagnose the faulty status without fusing the status from the neighboring sensor nodes. On the other hand, in the DSFDHT, DFD and IDFD algorithms, each sensor node requires 2, 5 and 3 messages respectively to identify the faulty sensor nodes. The number of messages exchanged in the network ($N = 512$) for all algorithms are tabulated in Table 5.8. From the table it has been found that the proposed DSFD3SET algorithm requires 50%, 66%, and 80% less message exchange overhead as compared to that of the DSFDHT, IDFD and DFD algorithms respectively.

5.5.4 Energy Consumption

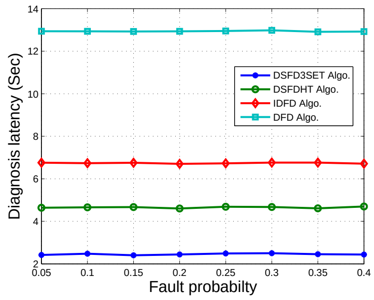
Total energy consumption depends on the number of message transmissions and receptions required for the diagnosis algorithm. The number of messages received



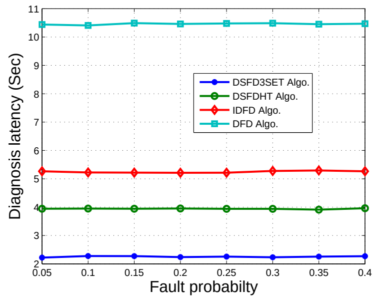
(a) For average degree $N_a = 10$



(b) For average degree $N_a = 15$

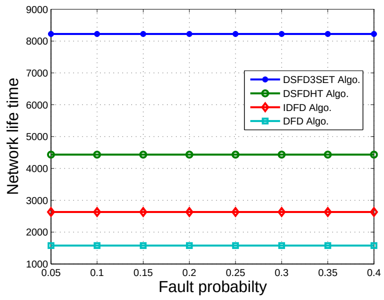


(c) For average degree $N_a = 20$

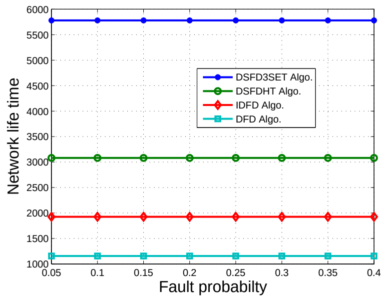


(d) For average degree $N_a = 25$

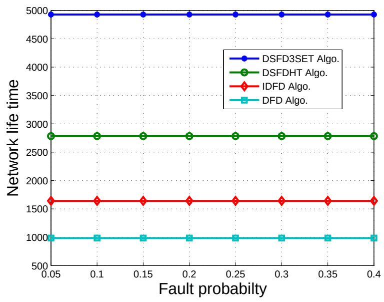
Figure 5.5: DL versus fault probability plots for the DSFD3SET, DSFDHT, DFD and IDFD algorithms.



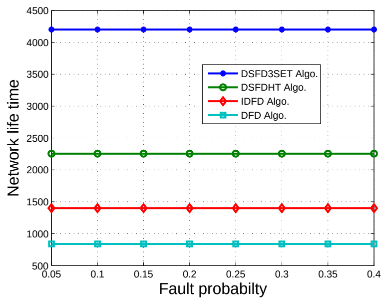
(a) For average degree $N_a = 10$



(b) For average degree $N_a = 15$



(c) For average degree $N_a = 20$



(d) For average degree $N_a = 25$

Figure 5.6: Network life time versus fault probability plots for the DSFD3SET, DSFDHT, DFD and IDFD algorithms.

depend on the number of sensor nodes in the network and the average degree N_a of the network. The total energy consumption of the algorithms with respect to different average degrees and varying fault probabilities are depicted in Figure 5.4 and Figure 5.7 respectively. The energy consumption in DSFD3SET is 45%, 51% and 77% less compared to that of DSFDHT, IDFD and DFD algorithms.

The energy consumption for all the algorithms increase when the average degree N_a of the network increases. It is because the number of message receptions increases when degree of a sensor node increases. Therefore, the DSFD3SET and other algorithms are linearly scalable. The fault diagnosis algorithm DSFD3SET is scalable due to the fact that the energy consumption increases slowly with respect to increase of the degree of the network, as compared to that of DSFDHT, DFD and IDFD algorithms. For large scale network, scalability is more important to preserve the energy consumption.

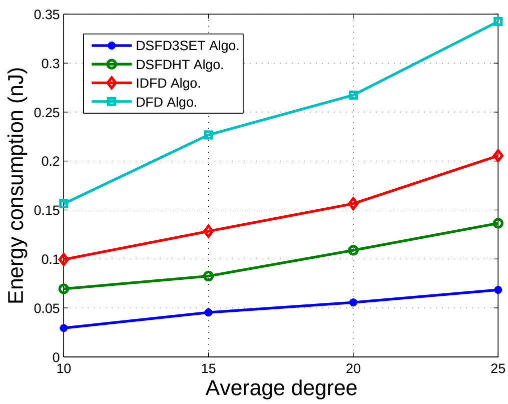


Figure 5.7: Energy consumption versus average degree N_a for the DSFD3SET, DSFDHT, DFD and IDFD algorithms

5.5.5 Diagnosis Latency (DL)

The diagnosis latency is the generic parameter used for evaluating the DSFD3SET algorithm which measures the time required to diagnose all the faulty sensor nodes in WSNs. The diagnosis latency versus fault probabilities for all the algorithms for different average degrees are depicted in Figure 5.5. The diagnosis latency of the DSFD3SET algorithm is improved by 40%, 78% and 57% with respect to the DSFDHT, DFD and IDFD algorithms respectively as shown in Figure 5.8.

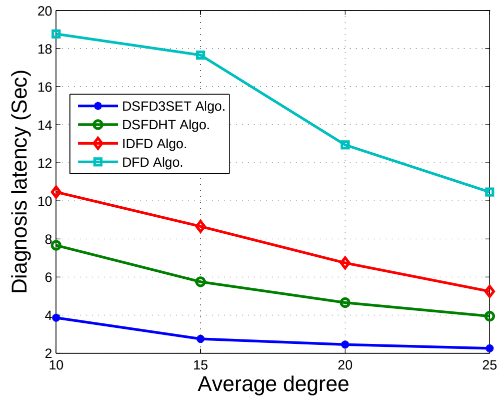


Figure 5.8: Diagnosis latency versus average degree for the DSFD3SET, DSFDHT, DFD and IDFD algorithms

5.5.6 Network Life Time

The network life time depends on the total number of messages exchanged over the network. As the communication overhead is less for the DSFD3SET algorithm as compared to that of the existing algorithms, the network life time of the DSFD3SET algorithm is improved by 24%, 80% and 68% over the DSFDHT, DFD and IDFD algorithms respectively as shown in Figure 5.6. The network life time of all the algorithms with respect to varying average degree and fault probabilities is shown in Figure 5.9 and Figure 5.6 respectively. Improvement of the DSFD3SET algorithm

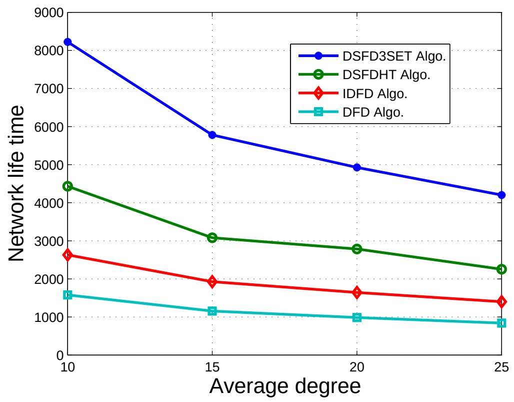


Figure 5.9: Network life time versus average degree N_a for the DSFD3SET, DSFDHT, DFD and IDFD algorithms

over DSFDHT, DFD and IDFD algorithms are tabulated in Table 5.9 for $N_a = 20$ and fault probability $P_f = 0.3$.

Table 5.9: Performance improvement of DSFD3SET algorithm over DSFDHT, DFD, IDFD algorithms when $N_a = 20$ and $P_f = 0.3$

Performance parameter	DSFD3SET Algorithm	DSFDHT Algorithm	DFD Algorithm	IDFD Algorithm	Improvements over DSFDHT Algorithm	Improvements over DFD Algorithm	Improvements over IDFD Algorithm
Diagnosis accuracy	0.98313	0.95636	0.92844	0.93697	3%	6%	5%
False alarm rate	0.0052	0.0541247	0.270833	0.240833	4%	26%	23%
False positive rate	0.0169	0.0436	0.0716	0.0631	2 %	5%	4%
Message exchange	512	1024	2560	1536	50%	80%	66%
Network life time	4927	2784	985	1642	43%	80%	66%
Eenergy consumption	0.029341	0.0696821	0.15670	0.0990231	45%	79%	61%
Diagnosis latency	2.49916	4.67551	12.9801	6.76327	46%	78%	57%

5.6 Conclusion

In this chapter, a modified three sigma edit test based distributed self fault diagnosis (DSFD3SET) algorithm for large scale WSNs is proposed using neighbor coordination. The performance of the DSFD3SET algorithm is compared with the existing algorithms and all the algorithms are simulated in NS3 simulator. The simulation results show that the proposed method outperforms over the existing algorithms by providing lower false alarm rate, false positive rate, high diagnosis accuracy, less diagnosis latency and more network life time. The diagnosis accuracy of the DSFD3SET algorithm is improved by 4%, 6% and 8% as compared to that of the existing DSFDHT, IDFD and DFD algorithms respectively when the average degree is 25. The DSFD3SET algorithm needs N message transmissions between the sensor nodes which is very less compared to the existing algorithms. Since less number of communications is needed to find the fault status, the algorithm is energy efficient and increases the lifetime of the network.

Chapter 6

Distributed Self Fault Diagnosis Algorithm to Diagnose Hard And Intermittent Faults in Large Scale WSN

Chapter 6

Distributed Self Fault Diagnosis Algorithm to Diagnose Hard and Intermittent Faults in Large Scale WSNs

In wireless sensor networks, the sensor nodes behave either fault free or faulty during different periods of time, which are considered to be intermittently faulty sensor nodes. The presence of intermittently faulty sensor nodes affect the network performance and fault diagnosis accuracy. Diagnosing intermittently faulty sensor nodes in wireless sensor networks is one of the important problems, because of inconsistent result patterns generated by the sensor nodes. The traditional distributed fault diagnosis algorithms consume more message exchanges to obtain the global fault status (i.e. status of all sensor nodes) of the network. To optimize the number of message exchanges over the network, the distributed self fault diagnosis is a preferable solution for WSNs as compared to traditional fault diagnosis. A self fault diagnosis algorithm is proposed here, which repeatedly conducts the self fault diagnosis procedure based on the modified three sigma edit test over duration to identify those intermittent faulty sensor nodes. The simulation results show that, the proposed DHISFD3SET algorithm has 12% improvements in diagnosis accuracy and 13% improvements in false alarm rate over the existing distributed intermittent fault diagnosis (DIFD) algorithm when 30% of sensor nodes are suffering from intermittent fault.

6.1 Introduction

Wireless sensor networks (WSNs) are effective and extensively used for providing the error free information based on the sensed values from the environment. The sensor nodes are more prone to become intermittently faulty and unreliable [104]. The presence of faulty sensor nodes in WSNs result in a significant performance deterioration. Therefore, it is necessary (some times also essential) to perform diagnosis timely, to find all kinds of abnormalities and fault condition in sensor networks. This will ensure the best quality of services (QoS) of WSNs.

A sensor node consists of different modules such as battery, sensor, microcontroller, transceiver, and memory, where each module performs a different task, such as supplying power, sensing, processing, send and receive, and storage respectively. When one or more modules become incorrect, it generates erroneous result which is known as a faulty sensor node and the presence of faulty sensor node leads to the failure of wireless sensor networks. Depending on the behavior of different modules in sensor node, the faulty sensor nodes are further classified into two types such as hard and soft faulty sensor nodes [43]. A hard faulty sensor node is unable to communicate with the rest of the sensor nodes present in the network. The main reason for a sensor node to be hard faulty is due to the defect in transceiver module (which is responsible for transmitting correct results to the rest of the sensor nodes present around a sensor node), processor module (as processor is the key point of functioning of all the modules present in a sensor node) and finally drainage of battery power from a sensor node [3, 81].

In the soft faulty sensor node, the transceiver, processor, and sensor module are working properly, however, due to either internal circuit damage or malicious attack, it generates erroneous results. Depending on the erroneous result pattern generation with respect to time, the soft faulty sensor node is further classified into transient and intermittent fault [88, 90]. The transient faulty sensor node [25] generates faulty results for only a single time, and it exists for short duration within the entire life span, whereas intermittently faulty sensor node generates faulty readings at different time instants [11].

Intermittent fault occurs in hardware system of the sensor node due to bad battery contacts, overheating of semiconductor ICs, noisy measurement from the sensors, and so forth. This fault also occurs in software systems as well. For example, exceptions and interrupts caused by some unknown bugs lead to crashes and reboots [3,25]. A number of fault diagnosis algorithms for WSNs are available in the literature [3,6,9,40,43,81]. These methods do not address the intermittently faulty sensor nodes. Many fault diagnosis algorithms are available in the literature to diagnose the intermittent faulty sensor nodes for a dynamic topology network [105,106]. A notion of failure diagnosability of discrete event systems was introduced by the authors Sampath *et al.* [107]. It follows a diagnosis procedure repeated for diagnosing the occurrence of a repeated number of faults in discrete event systems [108–110]. As WSN is suffering from battery constraint, low processing power, limited bandwidth, and low memory, the algorithms developed for multiprocessor and wired computer networks are not suitable.

Ssu *et al.* [90] have proposed a neighbor coordination based intermittent fault diagnosis algorithm in WSNs, where more number of iterations are required to find the intermittently faulty sensor nodes. Since each iteration needs message exchanges, more number of iterations are not suitable for WSNs. Distributed implementation of protocols for failure diagnosis of discrete event systems is reported by Debouk *et al.* [111]. A multi-objective optimization approach is adapted to find the parameters for intermittent fault detection in WSNs [112]. The recent work [25] focuses on the diagnosis of the number of occurrence of faults, however, fails to model the random intermittent fault behavior of sensor nodes. Our main concern is the robust diagnosis of current fault status, which reduces the number of tests required to diagnose the fault and proper modeling of the intermittent faults.

In this chapter, a robust and distributed intermittent self fault diagnosis (DH-ISFD3SET) algorithm is proposed. Modified three sigma edit test rule is applied repeatedly over the collected data by the sensor nodes to decide the current fault status. After the observed time expires, the sensor node decides its faulty behavior by diffusing all the decision information.

The major contributions of this chapter are

- Modeling of discrete intermittent fault events using Bernoulli distribution.
- Robust distributed technique to diagnose the current fault status of the sensor node based on the modified three sigma edit test methods.
- Energy efficient distributed self fault diagnosis procedure which reduces the number of tests required to detect the intermittent faults in WSNs.
- Implementation of the proposed algorithm in NS3 and demonstration of the efficiency by using standard parameters like diagnosis accuracy, false alarm rate, confidence interval (CI), and false positive rate .

The remaining part of this chapter is organized as follows. In Section 6.2, the network, fault, and data model used for the development of the algorithm are provided. The proposed DHISFD3SET algorithm is described in Section 6.3. The analysis of the new algorithm is given in Section 6.4. The simulation results are shown in Section 6.5. Finally, Section 6.6 concludes the chapter with discussions.

6.2 System Model

The system model consists of network and fault model proposed in Section 3.2. In network model, we specify the network topology and the way sensor nodes communicate each other. In fault model, the behavior of the sensor nodes on the occurrence of intermittent faults is presented.

6.2.1 Network Model

The network model considered here is same as discussed in Chapter 3. Let $S = \{s_1, s_2, \ldots, s_i, \ldots, s_N\}$ be a set of sensor nodes deployed in an environment of interest. If a sensor node is coming under the transmission range T_r of s_i at the time instant t then both the sensor nodes s_i and s_j are said to be connected. In the sensor network, only one hop communication between any pair of sensor nodes s_i and s_j is considered. Each of the sensor node s_i can communicate with its neighboring nodes $Neg_i(t) \subset S$. And the neighboring nodes $Neg_i(t)$ can also communicate

Table 6.1: The notations used for developing the DHISFD3SET algorithm

Symbol	**Description**
S	Set of sensor nodes in WSN.
C	Set of communication links among sensor nodes
s_i	A sensor node deployed at $P_i(xc_i, yc_i)$
N	Total number of sensor nodes deployed on the given terrine $R \times R$
NT_i	Neighboring table of the sensor node s_i containing all the information about its neighbors and itself.
$Neg_i(t)$	A set of neighboring sensor nodes of s_i at time instant t
$x_i(t)$	Modified sensed data of the sensor node s_i at the time instant t
A	Actual sensed data of the sensor node s_i
$w_i(t)$	Erroneous data sensed by the sensor node s_i
$fs_i(t)$	Fault status of the sensor node s_i at the time instant t
$FS_i(T)$	Fault status of the sensor node s_i calculated after the time duration T
T_r	Transmission range of sensor nodes
$\mathbf{Nx}_i(t)$	Set of Neighbor's data collected by s_i at the time instant t
S_G	A Set of fault free sensor nodes
S_F	A Set of faulty sensor nodes
S_1	A Set of hard faulty sensor nodes
S_2	A Set of intermittent faulty sensor nodes
p	Probability that a sensor node s_i is suffering with intermittent fault
α	Probability of a intermittent faulty sensor node produce wrong data
ζ	Minimum battery power at which a sensor node fails to work normally
T	Total observe time to diagnose the intermittent faulty sensor node
δ_T	The time duration after which another test will be done to study the intermittent behavior of the sensor node s_i
$MAD_i(t)$	Median absolute deviation over $Nx_i(t)$ at s_i
$MADN_i(t)$	Normalized median absolute deviation over $Nx_i(t)$ at s_i
N_i	Degree of the sensor node s_i
N_a	Average degree of the sensor nodes in the network
Re_i	Remaining energy of the sensor node s_i
T_I	Time at which the the self fault diagnosis procedure is started
θ	Threshold for computing the intermittently faulty sensor node

with it for which the sensor network is strongly connected. The data sensed by the sensor node s_i is stored locally on its memory and send it to their neighboring sensor nodes for testing. The average degree N_a of the sensor nodes depends on the T_r, i.e. if the T_r increases, then N_a also increases and vice verse. All the sensor nodes in WSNs communicate using wireless communication medium. Synchronous WSNs are assumed, where each sensor node sends and receives the messages from their neighboring nodes within a bounded time period. IEEE 802.15.4 is used as the MAC layer protocol for allowing the communication among the nodes.

6.2.2 Fault Model

Sensor nodes are subjected to both hard and soft faults. Let p is the probability that a sensor node is intermittently faulty. The set of randomly chosen sensor nodes ($\lceil pN \rceil$ numbers of faulty sensor node), which are subjected to either hard or soft fault (intermittent fault) is denoted as S_F and the set of fault free sensor nodes are denoted as S_G. The set S_F is further partitioned into two subsets S_1, and S_2. The set S_1 represents the set of hard faulty sensor nodes and S_2 represents the set of intermittently faulty sensor nodes in WSNs i. e. $S_F = S_1 \subset S_2$. The set of fault free sensor nodes $S_G = S - S_F$ and $|N| = |S| = |S_G + S_F|$. This shows that all the faulty and fault free sensor nodes in WSNs are included in set S. Also, it is assumed that $|S_F| << |S_G|$ i.e. the number of faulty sensor nodes are very less as compared to the number of fault free sensor nodes in WSNs.

Each sensor node s_i in WSNs is capable of sensing, transmitting, receiving, processing, forwarding, and taking a decision about its fault status based on their neighboring sensor nodes. In fact, these kinds of sensor nodes are counted as smart or intelligent sensor nodes. Each sensor node records the outcomes based on the neighboring node's observed data over time period T. During that time period, a sequence of outcomes for a sensor node s_i are identified under the following assumptions:

A1) The hard faulty sensor nodes are detected by using remaining battery power.

A2) The data of a sensor node s_i in each time instant t has two possible outcomes, i.e. either fault free or intermittent faulty.

A3) α is the probability that a sensor node s_i is intermittently faulty, i.e. the sensor node s_i fails to provide correct sensed data is α. The probability that a sensor node s_i provides correct data, i.e. a fault free sensor node's data is $1-\alpha$. This is modeled as the intermittent faulty behavior of a sensor node in WSN.

A4) The test outcomes are independent, i.e. the test outcome at one time instant has no influence over the outcome of another time instant.

The assumption A1 is used for diagnosing only hard faulty sensor node and the assumptions A2 through A4 are used for diagnosing the intermittently faulty sensor node. This process is modeled as the Bernoulli trials process which has a discrete distribution, having two possible outcomes labeled by $m = 0$ and $m = 1$. For $m = 1$, the fault occurs with probability α and for $m = 0$, the probability of failure of sensor node is $1-\alpha$. The probability density function is given as

$$f(m) = \alpha^m (1-\alpha)^{m-1} \tag{6.1}$$

The data of each sensor node s_i at time instant t, denoted as $x_i(t)$ is modeled by using the Bernoulli distribution of intermittent faults in successive measurements.

6.2.3 Data Model

Initially, all the sensor nodes are assumed to be fault free and charged with full battery power. Any of the sensor nodes is likely to suffer from an intermittent fault during their lifetime of deployment. It is assumed that the outcome $x_i(t)$ of s_i at time instant t depends on the true sensed value A of the unknown parameter and also based on random erroneous value in the observed data which is assumed to be additive. The sensor nodes are collecting data in regular interval δ_T for time duration T. The modified data (i.e. actual and erroneous value) is given as [95]

$$x_i(t) = A + w_i(t), \quad t = 1, 2, \ldots, K \quad K = \frac{T}{\delta T} \quad \text{and} \quad i = 1, 2, \ldots, N \tag{6.2}$$

where, $w_i(1), w_i(2), \ldots, w_i(K)$ are erroneous data at respective sensor nodes. It is a common assumption in the literature of WSNs that all the sensor nodes measurements have same mean with different erroneous data [100].

It is assumed that the random erroneous data are temporally and spatially independent and have the same distribution function F. It follows that the observation $x_i(1), x_i(2), \ldots, x_i(K)$ are independent with common distribution function and can say that the $x_i(t)$'s are *i.i.d.* i.e. independent and identically distributed. A conventional way to represent well-behaved data, i.e. data without fault, is assumed F is a normal distribution with mean A and variance σ_i^2 which implies $F = \mathcal{N}(A, \sigma_i^2)$.

The sensor node s_i suffered with an intermittent fault provides an arbitrary data for some time duration and behaves as a good sensor node in another time. In order to model this arbitrary behavior of the intermittently faulty sensor nodes, the Bernoulli distribution function is considered. Equation (6.1) and Equation (6.2) model the behavior of observed data from a sensor node s_i from the environment with the added erroneous data value for a particular time period T. In order to observe the intermittent faulty behavior, the value of $x_i(t)$ is given as

$$x_i(t) = [A + \upsilon_i(t)] + b(t)\upsilon_f(t) \tag{6.3}$$

where, $\upsilon_i(t)$ and $\upsilon_f(t)$ are independent zero mean Gaussian random variable with variances σ^2 and σ_f^2, respectively; $b(t)$ is a switch sequence of ones and zeros and is modeled as an independent and identically distributed Bernoulli random process with probability $P_r(b(t) = 1) = \alpha$ and $P_r(b(t) = 0) = 1 - \alpha$. The variance of $\upsilon_f(t)$ is chosen to be very large than that of $\upsilon_g(t)$ so that when $b(t) = 1$, a large error is experienced in $x_i(t)$. The $b(t)$ is given in Equation (6.4) as

$$b(t) = \begin{cases} 1, \ r \geq \alpha \\ 0, \ r < \alpha \end{cases} \tag{6.4}$$

where, r is a random variable in between 0 and 1. The corresponding pdf of the total error $(\upsilon_i(t) + b(t)\upsilon_f(t))$ in a sensor node suffered with intermittent fault in Equation (6.3) is given as

$$f(x) = \frac{1-\alpha}{\sqrt{2\pi}\sigma} \exp\left(\frac{(x-A)^2}{2\sigma^2}\right) + \frac{\alpha}{\sqrt{2\pi}\sigma_T} \exp\left(\frac{(x-A)^2}{2\sigma_T^2}\right) \tag{6.5}$$

where $\sigma_T^2 = \sigma^2 + \sigma_f^2$ and $E[|\upsilon_i(t) + b(t)\upsilon_f(t)|^2] = \sigma^2 + \alpha\sigma_f^2$. It is noted that when $\alpha = 0$ or 1, total error is a zero-mean Gaussian variate.

6.3 Distributed Self Fault Diagnosis Algorithm to Identify Intermittent Fault

Every sensor node s_i in the network is associated with K number of data which are measured at regular interval of time δT from its neighboring sensor nodes. The data for fault free and faulty sensor nodes is generated by using Equation (6.2) and Equation (6.3) respectively. Initially, it is assumed that as time elapses the faulty sensor node will generate αK number of faulty data at random instant of time as compared to the data sensed by fault free sensor nodes. The objective is to identify the faulty sensor nodes present in the network by analyzing the data of the different

Algorithm 6.1 DHISFD3SET Algorithm

```
Data: Observed time period T, sensed data x_i(t) at time t, intermittent fault probability (α), Battery
      power(Re_i) ζ, Initial time T_I
Result: Calculate S_1, S_2, and S_G at T
S_1 = ϕ , S_2 = ϕ, S_G = ϕ, n = T_I
if Re_i <= ζ then
 | S_1 = S_1 ∪ {s_i}
else
 | for n = T_I ··· (T_I+T)/δ_T do
 | | Each sensor node s_i collects environmental sensed data Nx_i(t) from their neighbors Neg_i(t).
 | | Sort(Nx_i(t))
 | | /* Procedure for sorting all the elements of Nx_i(t) in ascending order */
 | | if |Neg_i(t)|%2 == 0 then
 | | | md_i = [Nx_i(t)[|Neg_i(t)|/2] + Nx_i(t)[(|Neg_i(t)| + 1)/2]]/2
 | | else
 | | | md_i = [Nx_i(t)[|Neg_i(t)|/2]
 | | end
 | | ADM_i(t) = ϕ
 | | for j = 1 ··· |Neg_i(t)| + 1 do
 | | | ADM_i(t) = ADM_i(t) ∪ {(Nx_i(t)[j] − md_i)}
 | | end
 | | if |ADM_i(t)|%2 == 0 then
 | | | mad_i(t) = [ADM_i(t)[|Neg_i(t)|/2] + ADM_i(t)[(|Neg_i(t)| + 1)/2]]/2
 | | else
 | | | mad_i(t) = [ADM_i(t)[|Neg_i(t)|/2]
 | | end
 | | MADN_i(t) = mad_i(t)/0.675
 | | FSC_i(t) = (x_i(t) − md_i)/MADN_i(t)
 | | if FSC_i(t) < 3 then
 | | | FS_i(t) = 0
 | | else
 | | | FS_i(t) = 1
 | | end
 | end
 | if t == T + T_I then
 | | s = 0
 | | for k = T_I ··· (T+T_I)/δ_T do
 | | | s = s + FS_i(k)
 | | end
 | | if s < α(T/δ_T) then
 | | | S_G = S_G ∪ {s_i}
 | | else
 | | | S_2 = S_2 ∪ {s_i}
 | | end
 | end
end
```

sensor nodes in a distributed manner. If every sensor node shares its K number of observations from their neighboring nodes, then each sensor node keeps N_aK number of data, where N_a is the average degree of sensor nodes in the network. However, to avoid the storage problem of recording these large numbers of sensed data, now each sensor node s_i share data $x_i(t)$ to its neighboring nodes $Neg_i(t)$ in every cycle and predict the fault status at that time instant t. This process will continue for K times to identify the fault status by itself. Algorithm 6.1 depicts the distributed self fault diagnosis algorithm.

6.3.1 Hard Fault Diagnosis

During the life span, the battery power of a sensor node may be drained out and becomes unusable. To avoid such situation, a sensor node can detect the battery failure by periodical check-up of its energy level. If the energy level of a sensor node is less than the threshold value ζ, then that sensor node is considered to be hard faulty and does not send or receive any message from any other neighbors in WSNs. This is diagnosed by checking the remaining energy Re_i value of a sensor node s_i at repeated intervals of time t.

6.3.2 Intermittent Fault Diagnosis

The intermittently faulty sensor nodes are identified by measuring the outlyingness of an observation from the neighbors data. To make the algorithm robust, the modified three sigma edit test operation $f_i(t)$ is followed here, which is specified in Equation (6.10). In the proposed algorithm, each sensor node s_i measures the outlyingness present between its sensed data x_i with the estimated sensed data (which is calculated from the neighboring node's data $\mathbf{Nx}_i$ by using the Equation (6.10) and then compare the outlyingness $f_i(t)$ with a threshold θ. If the outlyingness exceeds the threshold θ then identify the sensor node s_i as faulty and keep the fault status in $FS_i(t)$. This procedure repeats for K times. Finally the intermittent fault status is computed by using Equation (6.12) which is discussed in Section 6.4. The detail description about the algorithm is discussed in Algorithm 6.1.

6.4 Analysis of the DHISFD3SET Algorithm

In this section, we analyze the proposed DHISFD3SET algorithm to estimate the performance and efficiency using standard generic performance parameters such as message complexity, diagnosis latency, and storage complexity. The robustness, correctness, and completeness are also proved based on the observed data analysis and using Lemma 6.1 through Lemma 6.5 and Theorem 6.1 and Theorem 6.2 respectively.

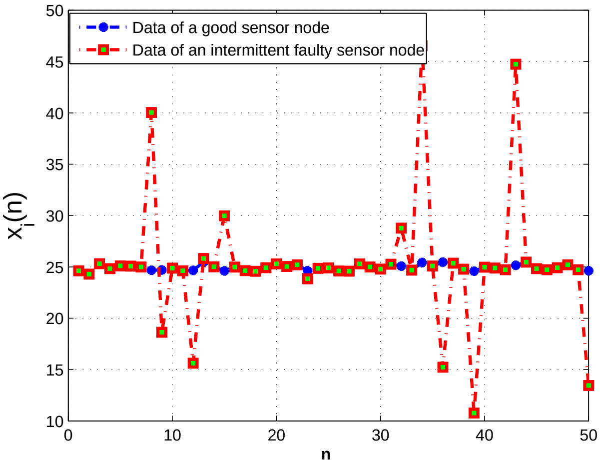


Figure 6.1: Behavior of an intermittent faulty sensor node where 20% of the time the sensor node fails to provide correct data. The true value is $A = 25$, the variances are $\sigma^2 = 0.1$, $\sigma_f^2 = 100$

6.4.1 Data Analysis

The data observed for characterizing the intermittent fault behavior of a sensor node is given in Equation (6.3). For instance, repeated measurements are taken based on the fact that 80% of the time it provides correct data and 20% of the time the sensor node fails to provide correct data. The data of a fault free and faulty sensor node are shown in Figure 6.1.

It has been seen from the Figure 6.1 that, the error may be large, but the average is approximated to the true value. At the same time, the confidence interval (CI) of the mean of the distribution may increase. The 95% CI of the mean for different values of time that a sensor node remains as intermittent fault is given in Table 6.2.

Table 6.2: Confidence interval

Fraction of suspicious data (α)	Confidence interval when different number of observations (K) taken from a sensor node	
	K=50	K=20
0 (fault free)	(24.90,25.10)	(24.85,25.13)
0.1	(24.08,25.92)	(23.75,26.26)
0.2	(23.43,26.50)	(23.09,26.97)
0.3	(23.02,26.93)	(22.49,27.39)
0.4	(22.91,27.22)	(22.17,27.85)
0.5	(22.70,27.38)	(21.88,28.24)

From the Table 6.2, it is observed that the CI is (24.90, 25.10) for a fault free sensor node. It increases when the percentage of erroneous data sensed by a faulty sensor node increases. The mean is not used as a parameter to compare with true value in order to detect the intermittent faulty sensor node. Moreover, in distributed case, a sensor node collects data from the neighbors and predicts the fault status. To do this, the sensor nodes need to keep all the data from the neighboring nodes which needs large memory to store them [25].

In order to reduce the storage requirement, few data from the neighboring nodes are stored. However, from the Table 6.2, it is evident that when the number of data points are less, the CI is more. Therefore, the method of comparing mean will not provide accurate solutions to diagnose the intermittent faults unlike in most of the conventional fault diagnosis algorithms which are based on comparison of own data with the neighboring node's data [62] or mean of the neighbors and its own data [39]. In order to improve the reliability of the results, in this work, we have adapted, modified three sigma edit test to diagnose the intermittently faulty sensor nodes, which computes absolute error in the data and status. Instead of storing all the data from neighboring nodes, the sensor node s_i only stores the absolute error (6.10) $f_i(t)$ in its memory. As sensor nodes are memory constrained, the storage required is reduced in the proposed algorithm.

6.4.2 Analysis of the DHISFD3SET Algorithm

Each sensor node accumulates the data from the neighboring nodes $Neg_i(t)$ at time instant t and stores in $\mathbf{Nx}_i(t) = \{x_i(t)\}_{s_i \in Neg_i(t)}$. The outlyingness is measured taking both estimated mean $\hat{\mu}_i(t)$ and standard deviation (SD) of the data collected from the neighboring nodes. The standard deviation $\hat{\sigma}_i(t)$ at sensor node s_i is defined as

$$\hat{\sigma}_i(t) = \sqrt{\frac{1}{N_i - 1} \sum_{s_j \in Neg_i(t)} (x_j(t) - \hat{\mu}_i(t))^2} \tag{6.6}$$

The outlyingness $t_i(t)$ is the ratio between its deviation to the estimated mean $\hat{\mu}_i(t)$ and SD $\hat{\sigma}_i(t)$. This is calculated as

$$t_i(t) = \frac{x_i(t) - \hat{\mu}_i(t)}{\hat{\sigma}_i(t)} \tag{6.7}$$

According to the 'three-sigma-edit' rule, a sensor node is regarded as faulty if $|t_i(t)| > \theta$. Otherwise, the sensor node is considered as fault free. The conventional three-sigma edit rule is ineffective for a small number of samples. According to the statistical feature, if $N_i < 10$, then $|t_i(t)|$ is always less than 3 with the CI of 95% where N_i is the degree of sensor node s_i [95]. This shows that the rule is ineffective for the lower average degree network. This measure is used to identify the presence of a faulty or fault free sensor node in WSNs. This measure is also unable to track, if two sensor nodes data are erroneous and one faulty sensor node data is very large compared to that of another faulty sensor node data. In this situation, the faulty sensor node may become fault free. This effect is called *masking*. The ineffectiveness of this measure is due to non robust nature of mean and SD. The estimated mean and SD deviates more when a faulty sensor node present in the neighborhood.

In order to overcome this problem, in literature, median of data has been used instead of the mean. The median of the data set $\mathbf{Nx}_i(t) = \{x_1(t), x_2(t), ..., x_{N_i}(t)\}$ is calculated after sorting the observation in increasing order as

$$x_1(t) \leq x_2(t) ... \leq x_{N_i}(t)$$

If N_i is odd (i.e. $N_i = 2m - 1$ for some integer m), then the median $Md_i(t) = \mathrm{Med}(\mathbf{Nx}_i(t)) = x_m(t)$ If N_i is even (i.e. $N_i = 2m$ for some integer m), then the median is defined as

$$Md_i(t) = Med(\mathbf{Nx}_i(t)) = \frac{x_m(t) + x_{m+1}(t)}{2}$$

Similarly, another alternative to the SD is the median of the absolute deviation (around the mean) of the observation from the median and it is known as median absolute deviation (MAD) [102]. This is defined as

$$MAD(x_1(t), ..., x_{N_i}(t)) = \mathrm{Med}|x_i(t) - Md_i(t)| \tag{6.8}$$

Assuming a normal distribution, it is observed that $MAD(x_1(t), ..., x_{N_i}(t)) = 0.675SD$. Therefore, in order to use the MAD as like SD, the normalized median absolute deviation (about the median) $\mathrm{MADN}(x_i(t))$ is used which is defined as

$$\mathrm{MADN}(\mathbf{Nx}_i(t)) = \frac{\mathrm{Med}\{|x_i(t) - Md_i(t))|\}}{0.675} \tag{6.9}$$

It is also observed that the Med($\mathbf{Nx}_i(t)$) and MADN($\mathbf{Nx}_i(t)$) are robust compared to that of $\hat{\mu}(t)$ and SD, when data is contaminated with outliers which is generated from the unknown mechanisms of the faulty sensor node. To avoid the drawback of the above fault diagnosis method based on $\hat{\mu}$ and SD, the mean is replaced by the median of the neighbor's data Med($\mathbf{Nx}_i(t)$). In the place of SD, the normalized median absolute deviation about the median (MADN($\mathbf{Nx}_i(t)$)) is used. The new measure of outlyingness $f_i(t)$ to detect the fault status of a sensor node is given by

$$f_i(t) = \frac{x_i(t) - Md_i(t)}{\text{MADN}(\mathbf{Nx}_i(t))} \tag{6.10}$$

where $f_i(t)$ is the absolute error for the modified three sigma edit test.

This method is accurate when the number of faulty sensor nodes present in the neighbor is more. The modified three sigma edit test operation $f_i(t)$ given in Equation (6.10) is performed and the fault status of a sensor node is identified as

$$fs_i(f_i(t)) = \begin{cases} 1, \; f_i(t) \geq \theta \\ 0, \; f_i(t) < \theta \end{cases} \tag{6.11}$$

Where θ is threshold.

This process is repeated for $K = \left\lceil \frac{T}{\delta T} \right\rceil$ times and the fault status for different consecutive instances are stored in $\mathbf{fs}_i$. Each sensor node s_i establishes its own intermittent fault status (faulty or fault free) at the end of K iterations by using Equation (6.12) is defined as

$$FS_i(T) = \begin{cases} 1, \; \sum_{k=1}^{K} fs(f_i(k)) \geq \lceil \alpha K \rceil \\ \\ 0, \; \sum_{k=1}^{K} fs(f_i(k)) < \lceil \alpha K \rceil \end{cases} \tag{6.12}$$

where, k is an integer between 1 to K and $i = 1, 2, ..., N$.

In order to ensure the robustness, correctness and completeness of DHISFD3SET algorithm, the following lemmas and theorms are given as follows.

Lemma 6.1

The proposed self fault diagnosis algorithm DHISFD3SET is robust.

Proof

At time instant t, each sensor node has own data $x_i(t)$ and data from the neighbors $Nx_i(t)$. Though several methods are available in literature, in one approach, the sensor node s_i compares own data with each of the neighbor's data. If the difference is more than a certain threshold value (which is common to all sensor nodes), then s_i considered s_j as probable faulty [62]. This is given by

$$|x_i(t) - x_j(t)| < \gamma_1 \tag{6.13}$$

where $i = 1 \cdots N$ and $j \in Neg_i$. This approach may provide incorrect result when both the sensor nodes are faulty. Because, the algorithms detect two faulty sensor nodes as fault free, if the variation of the data between them is less. Both of them erroneously detected themselves as fault free.

If the sensor node compares the received data with the true value A as given in Equation (6.14) and absolute value of difference between actual value A and $x_i(t)$ is less than γ_2, then

$$|A - x_j(t)| < \gamma_2 \tag{6.14}$$

the sensor node s_i predicts the probable fault status of s_j is faulty [10, 39]. This method is applied where the true value A is known. In fact, the estimated mean can be used instead of actual data A. The sample mean $\hat{\mu}_i$ at sensor node s_i is defined in Equation (6.15) as,

$$\hat{\mu}_i = \frac{1}{N_i} \sum_{s_j \in Neg_i} x_j \tag{6.15}$$

where N_i is the degree of s_i.

A statistical measure of the outlyingness of an observation of a sensor node x_i with respect to an estimated mean $\hat{\mu}_i$ is defined in Equation 6.16 [39] as

$$d_i = |x_i - \hat{\mu}_i| \tag{6.16}$$

where, d_i is the deviation between x_i and $\hat{\mu}_i$ of the sensor node s_i. The sensor node itself is identified as faulty if $d_i > \gamma_2$ (threshold) otherwise it is fault free. When a fault occurs, the estimated mean deviates more which affect to a fault free sensor node detected as faulty. This ensures that the algorithm is robust as compared to the method where the difference between the observed data for any pair of neighboring sensor nodes is comparing with the threshold. This proves Lemma 6.1.

Lemma 6.2: The diagnosis latency (DL) of the DHISFD3SET algorithm is $O(K \times T_{out} + T_{proc})$, where T_{out} is the maximum time set by the sensor node when the message exchange occurs in any pair of sensor nodes, T_{proc} is the maximum time required by the algorithm for processing, and K is the total number of times the data is received from the neighboring nodes.

Proof

The DL of the DHISFD3SET algorithm is the total amount of time required to diagnose all faulty sensor nodes in the network. In the communication graph $G = V(S, C)$ of the wireless sensor network, each sensor node communicates with the one hop neighboring sensor nodes only. Let T_{out} is the maximum time set by the timer when the message exchange occurs among the sensor nodes. The DHISFD3SET algorithm exchanges K number of messages. Therefore, the total time needed for message exchange is KT_{out}. Let T_{proc} is the maximum time required by the algorithm for processing both sensed data and computing fault status.

The total time required by the DHISFD3SET algorithm to diagnose all the faulty sensor nodes is $T_{DHISFD3SET} = O(K \times T_{out} + T_{proc})$. The self fault diagnosis algorithm achieves the diagnosis within a bounded delay of $T_{DHISFD3SET}$, due to a synchronous WSN as specified in network model of Section 6.2. This proves Lemma 6.2.

Lemma 6.3: The message complexity of the DHISFD3SET algorithm is $O(K \times N)$ where N is the number of sensor nodes in WSN and K is the total number of iterations required to judge the intermittent fault behavior of the sensor node.

Proof

The message complexity is the total number of messages exchanged over the network to get the final fault status of all the sensor nodes in the network. In DHISFD3SET, each sensor node s_i sends the sensed data to its neighbors, costing one message per node i.e. N messages in the network. This procedure is repeated for K times to identify the final fault status of a sensor node. The DHISFD3SET algorithm exchanges at most KN messages for self fault diagnosis, where N is the total number of sensor nodes in the network and K is the total number of times each sensor node s_i needs data from their neighbors Neg_i to judge the final fault status. Therefore, the total number of messages exchanged for the DHISFD3SET algorithm is $M_{DHISFD3SET}$ given as.

$$M_{DHISFD3SET} = O(K \times N) \tag{6.17}$$

This proves Lemma 6.3.

Lemma 6.4: The storage complexity of the DHISFD3SET algorithm is $O(K)$ where K is the total number of iterations required to identify the intermittent behavior of the sensor nodes.

Proof

In the DHISFD3SET algorithm, each sensor node s_i keeps the current diagnosed fault status information based on the absolute error between own sensed data and normalized median of neighboring node's data $\mathbf{Nx}_i$. If the absolute error $f_i(t)$ is less than θ then the sensor node is recorded as fault free otherwise faulty. Only one bit of information, i.e. 1 or 0 to represent fault free or faulty status of a sensor node s_i is stored. As this procedure is repeated for K times, each sensor node s_i needs only K bits. Along with this it also needs another 1 bit of information for storing the final fault status. Therefore, the total storage required by a sensor node s_i to keep all the required information is $K + 1$ which is $O(K)$. This proves Lemma 6.4.

Theorem 6.1: DHISFD3SET algorithm finds all faulty nodes present in the network correctly.

Proof

According to the diagnosis literature [96, 97], an algorithm is said to be correct, if a faulty sensor node is diagnosed as faulty with better diagnosis accuracy which is defined in Section 6.5. In order to prove the correctness property, we have to initially ensure that every sensor node is connected with their neighbor so that the message exchange between neighboring nodes is possible. This is ensured by the connectedness of every sensor node with their neighboring nodes.

In order to prove the connectedness property, we consider the parameters such as transmission range T_r, average degree N_a and the modified three sigma edit test rule applied over the own sensed data x_i and the observed data $\mathbf{Nx}_i$ of neighbors. The DHISFD3SET algorithm performs the diagnosis on each sensor node s_i of the sensor network. As a sensor node s_i communicates with their neighbors which are coming within its T_r, the correctness property of WSNs ensures that each and every sensor node s_i coming under the transmission range of one or more sensor nodes in the network. Therefore the diagnosis of each sensor node s_i is achieved.

According to the DHISFD3SET algorithm, every sensor node s_i collects the observed data from its neighboring node Neg_i and compute the absolute error $f_i(t)$ between its data and neighbor's data at the time instant t. If $f_i(t)$ is less than θ, then the sensor node s_i is diagnosed as fault free because the absolute error is less. Otherwise, the sensor node is diagnosed as faulty as the deviation between the observed data of sensor node s_i is more as compared to the data collected from the neighboring nodes. This process is iterated for K times so that the fault status of K different, but consecutive time instant is obtained. Out of K time instants, if $K/2$ or more than $K/2$ times, the fault status is reported, it is considered that sensor node s_i is intermittently faulty. When the sensor node depletes its battery to power below the threshold ζ, the sensor node s_i is considered to be hard faulty.

As the diagnosis process depends on the absolute error between the data of sensor node s_i and neighboring sensor nodes Neg_i, the sensor node s_i may be misdiagnosed due to the fact that the fault free neighboring sensor node provide correct data i.e. claim to be fault free. In order to prevent this misdiagnosis, the DHISFD3SET algorithm consider K iteration assuming that if the number of iterations increases, the probability of error decreases. Because, a fault free sensor node provide correct data maximum times, whereas a faulty sensor node provide erroneous thereby fault free sensor node is diagnosed as fault free and faulty sensor node is diagnosed as faulty at the end of K number of iterations. This ensures the correctness of diagnosis for hard fault, intermittent fault, faulty sensor node clamming fault free (false positive rate), and fault free sensor node claim to be faulty (false alarm rate). Thereby, the proposed DHISFD3SET algorithm is correct. This proves Theorem 6.1.

Theorem 6.2: The proposed DHISFD3SET algorithm is complete.

Proof

The algorithm is said to be complete, if no sensor node remains undiagnosed at the end of the diagnosis process. As stated in Section 6.2, the network is connected. We have to prove that each sensor node needs to be diagnosed under the faults such as hard fault, intermittent fault, faulty sensor node clamming fault free (false positive rate), and fault free sensor node claim to be faulty (false alarm rate). In the Theorem 6.1, we have proved that the hard fault, intermittent fault, faulty sensor node clamming fault free (false positive rate), and fault free sensor node claim to be faulty is diagnosed correctly. Due to the connectedness of WSN, no sensor node remains undiagnosed because, each sensor node has a path to another sensor node. Many times diagnosis procedure depends only on the communication between sensor nodes and their neighboring nodes in order to diagnose the hard fault, intermittent fault, faulty sensor node clamming fault free, and fault free sensor node claim to be faulty. This proves Theorem 6.2.

Table 6.3: Simulation parameters

Parameter	**Value**
Number of sensor nodes(N)	1024
Simulation time	$300s$
Propagation loss model	Range propagation loss model
Coverage area	$1000m \times 1000m$
Fault model	Normal Random Variable
Transmission range(T_r)	$(56, 61, 68, 74))cm$
Network type	Homogeneous
Node mobility	Constant speed mobility model
Environment condition	variation in environment and noise is considered
Node distribution	Uniform random distribution
Node capacity	5 buffers for receiving packets
Sensed data of fault free and faulty sensor node	Normal random variable with mean(μ) 30 and variance(σ) 1 and 1000 for fault free and faulty sensor nodes respectively

6.5 Result and Discussion

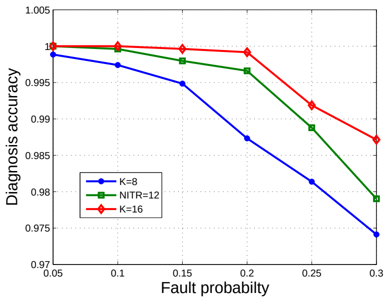
In order to evaluate the performance of the proposed DHISFD3SET algorithm the network simulator NS3 (version 3.19) [38] is used considering a discrete event network simulation. The performance of the DHISFD3SET algorithm is compared with existing DIFD algorithm [25] to validate the result. The network parameters used in the simulation are provided in Table 6.3. The performance of the algorithms is measured in terms of diagnosis accuracy, false alarm rate and false positive rate which are defined in Chapter 3.

6.5.1 Simulation Model

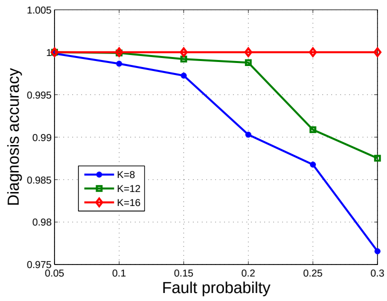
The DHISFD3SET algorithm is tested for different fault probabilities from 0.05 to 0.3 in the step size of 0.05. Since the statistical method's performance depends on the degree of the network, the algorithm is verified for different average degrees N_a, which is represented as a graph consisting of a set of vertices and set of edges. In order to obtain the average degree of a sensor node from 10 to 25 with step size of 5, the transmission ranges are chosen as $56, 61, 68$, and 74 respectively. We have performed 100 experiments for each point in the graph and average is plotted. The simulation results show that the proposed method outperforms DIFD and also it is observed that if a sensor node suffers from intermittent faulty for a long duration, identifying its fault status is reliable. The algorithm is tested for different intermittent fault probabilities which range from 0.6 to 0.9 with the step size of 0.1.

6.5.2 Estimation of the Minimum Number of Testing Required to Diagnose the Intermittent Fault

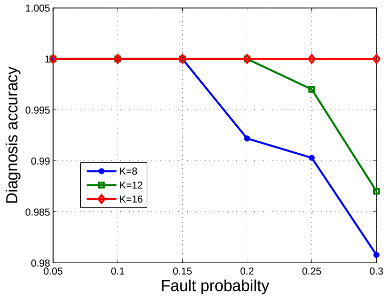
The minimum number of testing required for identifying the intermittently faulty sensor node is initially estimated. As the behavior of the intermittently faulty sensor node changes from one time instant to another, the DHISFD3SET algorithm is executed for minimum number of iterations (K) to achieve diagnosis. The result shows that the diagnosis accuracy is 100% when the number of iterations i.e. value of K is 16 with respect to 95% CI.



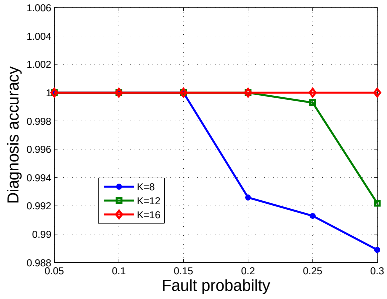
(a) For average degree $N_a = 10$



(b) For average degree $N_a = 15$

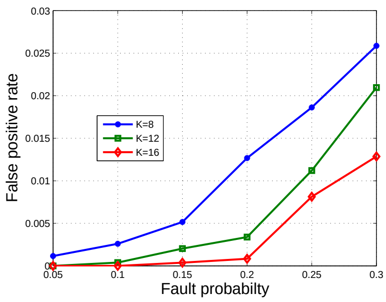


(c) For average degree $N_a = 20$

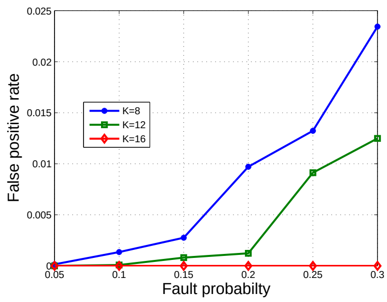


(d) For average degree $N_a = 25$

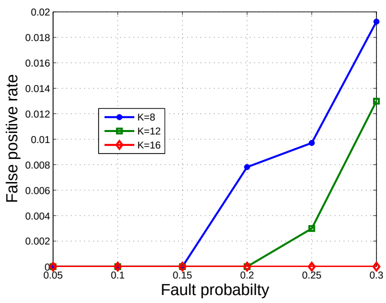
Figure 6.2: Diagnosis accuracy versus fault probability plots for the DHISFD3SET algorithm.



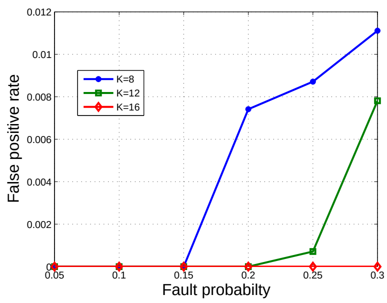
(a) For average degree $N_a = 10$



(b) For average degree $N_a = 15$

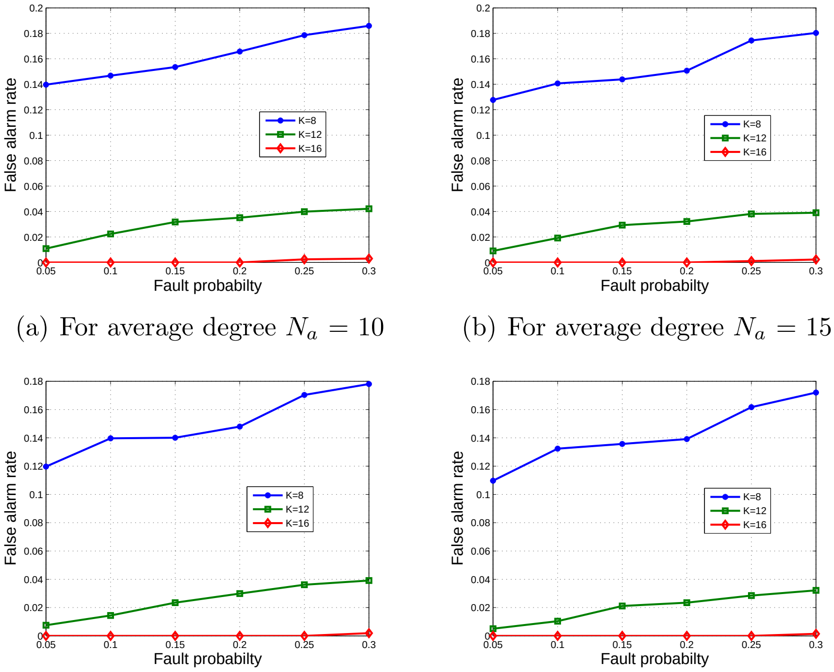


(c) For average degree $N_a = 20$



(d) For average degree $N_a = 25$

Figure 6.3: False positive rate versus fault probability plots for the DHISFD3SET algorithm.



(a) For average degree $N_a = 10$

(b) For average degree $N_a = 15$

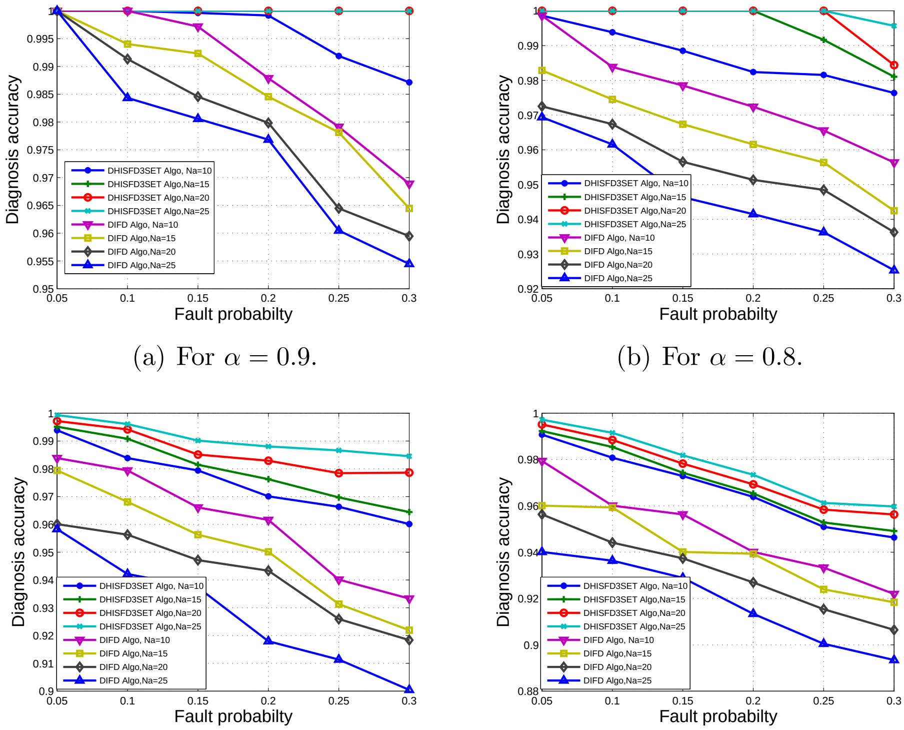
(c) For average degree $N_a = 20$

(d) For average degree $N_a = 25$

Figure 6.4: False alarm rate versus fault probability plots for the DHISFD3SET algorithm.

In the simulation, the faulty nature of each sensor node is observed for the time duration of $T = 300s$. Over this time duration, the fault status of each sensor node is tested for 8, 12, and 16 times by choosing the time interval δT as $37s$, $25s$ and $19s$ respectively. The data for an intermittently faulty sensor node is generated by using a Bernoulli distribution which is given in Section 6.2.3.

The diagnosis accuracy, false positive rate and false alarm rate performances for different average degrees are shown in Figure 6.2, Figure 6.3, and Figure 6.4 respectively. The result shows that, the DHISFD3SET algorithm gives 100% diagnosis accuracy, 0% false alarm rate, and 0% false positive rate when the number of testing iterations is 16 for the fault probability of $p = 30\%$ with average degree 25 and intermittently fault probability of $\alpha = 90\%$. The minimum 16 number of testing iterations are required to identify the intermittent faulty sensor node. Whereas, the DIFD algorithm [25] needs 21 iterations for achieving the same level of performance. Therefore, the proposed algorithm saves 33% of the energy of the sensor node which can be utilized for normal workloads of the sensor network. The DHISFD3SET algorithm needs less iterations to achieve diagnosis. The proposed algorithm, model the fault behavior using Bernoulli distribution and modified three sigma edit test method to achieve a diagnosis. On the other hand the DIFD algorithm uses some random data and neighbor coordination approach for diagnosis.



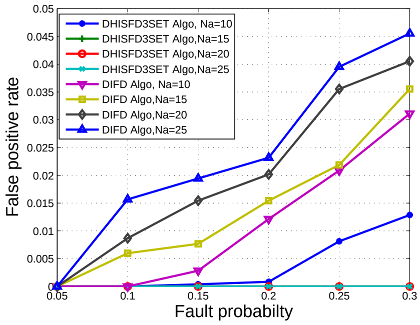
(a) For $\alpha = 0.9$. (b) For $\alpha = 0.8$.

(c) For $\alpha = 0.7$. (d) For $\alpha = 0.6$.

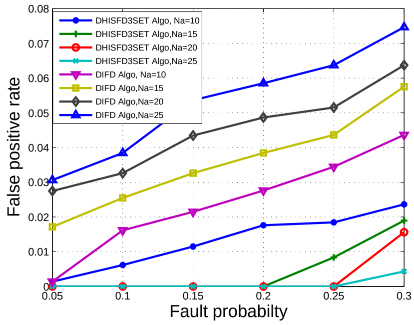
Figure 6.5: Diagnosis accuracy versus fault probability plots of the DHISFD3SET and DIFD algorithms for different N_a and α.

6.5.3 The diagnosis accuracy, false alarm rate and false positive rate Performance

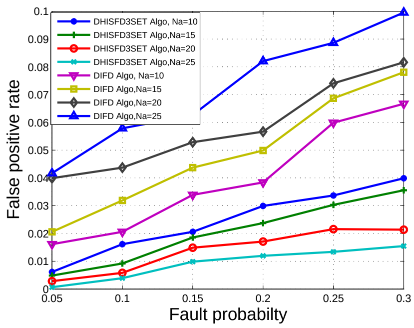
After estimating the minimum number of testing iterations required to identify the intermittently faulty sensor nodes in the worst scenario, the efficiency of the DHISFD3SET algorithm is tested for different intermittent fault probabilities (α). When an intermittent faulty sensor node provides erroneous data for longer duration, identifying intermittent faulty sensor nodes is reliable with high probability. However, difficulty arises when a sensor node's sensed data is suspicious for less duration. The robustness of the algorithm is verified for different α values and observed that the algorithms performance degrades if the intermittent fault probability of a sensor node is less than 0.6.



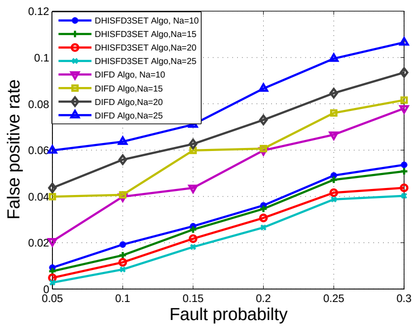
(a) For $\alpha = 0.9$.



(b) For $\alpha = 0.8$.



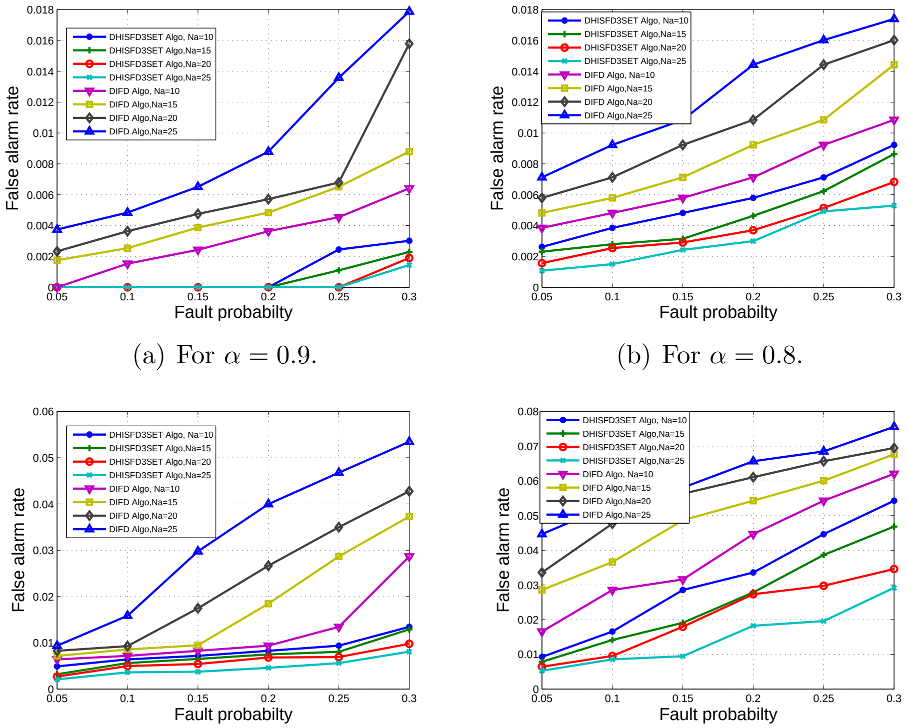
(c) For $\alpha = 0.7$.



(d) For $\alpha = 0.6$.

Figure 6.6: False positive rate versus fault probability plots of the DHISFD3SET and DIFD algorithms for different N_a and α.

The diagnosis accuracy, False positive rate and false alarm rate performances of the DHISFD3SET algorithm for different fault probabilities p and by varying α and average degree N_a are given in Figure 6.5, Figure 6.6 and Figure 6.7 respectively. The results are compared with existing DIFD algorithm [25]. The proposed scheme gives nearly 90% diagnosis accuracy, 10% false positive rate and 7% false alarm rate for intermittent fault probability of $\alpha = 0.6\%$. The diagnosis algorithm DHISFD3SET give improvement of 6% in diagnosis accuracy, 7% in false positive rate, and 5% in false alarm rate over DIFD algorithm, when intermittent fault probability (α) is $\alpha = 0.7$, the average degree of the network (N_a) is 25 and network size is 1024. The comparison results are shown in Figure 6.5, Figure 6.6 and Figure 6.7. This is due to the fact that in the DHISFD3SET algorithm the fault status is observed by a robust statistical method. The robust method is to identify the faulty sensor node more accurately compared to the neighbor coordination method used in DIFD algorithm [25].



(a) For $\alpha = 0.9$.

(b) For $\alpha = 0.8$.

(c) For $\alpha = 0.7$.

(d) For $\alpha = 0.6$.

Figure 6.7: False alarm rate versus fault probability plots of the DHISFD3SET and DIFD algorithms for different N_a and α.

6.5.4 Result Analysis with Respect to Confidence Interval

The 95% confidence interval (CI) of diagnosis accuracy and false alarm rate for different fault probabilities (p), intermittent fault probabilities (α) and average degrees of sensor nodes (N_a) are provided in Table 6.4 and Table 6.5 respectively. From the tables, it is shown that the CI is less for the DHISFD3SET algorithm as compared to that of the DIFD algorithm, with respect to different fault probabilities and intermittent data fault probability. The CI increases when the fault probability of the network increases for both the algorithms. However, the CI decreases when α increases for constant p. It is because when α increases, the faulty sensor node provides inconsistent data more frequently. This helps the fault detector to detect the intermittent fault behavior. Similarly, the CI decreases when the average degree of the network increases in the proposed algorithm. Whereas, in the DIFD algorithm, the performance decreases, i.e. CI increases when the degree of sensor node increases. This is due to the neighbor coordination method for fault diagnosis.

Table 6.4: Confidence interval of diagnosis accuracy for the DHISFD3SET (Algo 1), and DIFD (Algo 2) algorithms

IFP	FP	CI when Na = 10		CI when Na = 15		CI when Na = 20		CI when Na = 25	
		Algo 1	Algo 2	Algo 1	Algo 2	Algo 1	Algo 2	Algo 1	Algo 2
	0.05	100.00, 100.00	93.18, 100.00	100.00, 100.00	93.19, 100.00	100.00, 100.00	89.67, 100.00	100.00, 100.00	87.10, 100.00
	0.10	96.27, 100.00	93.71, 100.00	97.08, 100.00	92.22, 99.78	97.08, 100.00	90.79, 99.29	97.08, 100.00	89.41, 98.58
	0.15	94.31, 99.69	92.90, 99.10	94.33, 99.69	91.53, 98.44	95.79, 100.00	90.25, 97.75	95.79, 100.00	90.25, 97.75
0.6	0.2	94.66, 99.34	92.03, 97.94	94.66, 99.34	90.75, 97.25	94.66, 99.34	89.52, 96.48	96.08, 99.92	88.29, 95.71
	0.25	92.33, 97.66	89.87, 96.13	92.33, 97.67	89.17, 96.68	93.60, 98.40	88.68, 95.33	93.60, 98.40	86.33, 93.68
	0.3	92.57, 97.43	88.97, 95.03	92.57, 97.43	88.97, 95.18	93.81, 98.19	87.87, 94.13	93.81, 98.19	86.65, 93.35
	0.05	100.00, 100.00	100.00, 100.00	100.00, 100.00	93.09, 100.00	100.00, 100.00	93.99, 100.00	100.00, 100.00	89.67, 100.00
	0.1	97.47, 100.00	94.94, 100.00	100.00, 100.00	93.67, 100.00	100.00, 100.00	92.27, 99.78	100.00, 100.00	90.79, 99.21
	0.15	95.79, 100.00	94.35, 99.69	97.43, 100.00	92.93, 99.19	97.43, 100.00	91.56, 98.44	97.43, 100.00	90.25, 97.72
0.7	0.2	94.66, 99.34	93.93, 99.34	95.98, 99.92	92.07, 97.98	97.64, 100.00	92.02, 97.78	97.64, 100.00	88.29, 95.71
	0.25	94.95, 99.09	91.17, 96.82	94.85, 99.09	89.87, 96.13	96.29, 99.72	89.87, 96.13	97.78, 100.00	87.49, 94.51
	0.3	93.81, 98.19	90.51, 96.48	95.09, 98.91	88.97, 95.08	96.44, 99.56	88.97, 95.03	97.89, 100.00	86.65, 93.35
	0.05	100.00, 100.00	100.00, 100.00	100.00, 100.00	100.00, 100.00	100.00, 100.00	93.99, 100.00	100.00, 100.00	94.19, 100.00
	0.1	100.00, 100.00	96.47, 100.00	100.00, 100.00	95.44, 100.00	100.00, 100.00	93.57, 100.00	100.00, 100.00	92.27, 100.00
	0.15	97.43, 100.00	95.59, 100.00	100.00, 100.00	94.45, 99.69	100.00, 100.00	92.93, 99.09	100.00, 100.00	91.47, 99.52
0.8	0.20	97.20, 100.00	96.08, 99.91	100.00, 100.00	93.63, 99.33	100.00, 100.00	92.47, 97.98	100.00, 100.00	90.07, 98.44
	0.25	95.74, 99.71	94.88, 99.07	97.89 , 100.00	92.84, 98.48	100.00, 100.00	92.33, 97.66	100.00, 100.00	91.07, 96.90
	0.3	96.44, 99.56	93.82, 98.17	96.44, 99.56	91.52, 96.65	97.88, 100.00	91.51, 96.68	100.00, 100.00	89.95, 96.74

Similarly, the analysis of CI of false alarm rate is given here. The range of values between the minimum and maximum of false alarm rate in percentage when $p = 0.3$, $\alpha = 0.6$ and $N_a = 25$ is 1.26 to 3.49 and 5.73 to 9.63 with respect to CI of 95% for the DHISFD3SET and DIFD algorithms respectively. Similarly, The range of values between the minimum and maximum of diagnosis accuracy in percentage when $p = 0.3$, $\alpha = 0.6$ and $N_a = 25$ is 93.81 to 98.19 and 86.65 to 93.35 with respect to a confidence interval of 95% for the DHISFD3SET and DIFD algorithms respectively. This clearly shows that the proposed algorithm outperforms over the existing algorithms. Even though, when $\alpha = 0.8$ and $N_a = 25$, the diagnosis accuracy is 100%, but the DIFD algorithm performance is not satisfactory. The CI is (89.95, 96.74). Therefore, from the tables it has been seen that the CI performance of the

Table 6.5: Confidence interval of false alarm rate for the DHISFD3SET (Algo 1), and DIFD (Algo 2) algorithms

IFP	FP	CI when Na = 10		CI when Na = 15		CI when Na = 20		CI when Na = 25	
		Algo 1	Algo 2	Algo 1	Algo 2	Algo 1	Algo 2	Algo 1	Algo 2
	0.05	0.39, 1.66	0.92, 2.57	0.32, 1.53	1.83, 3.93	0.19, 1.25	2.26, 4.53	0.06, 0.96	3.22, 5.83
	0.1	0.89, 2.58	1.84, 4.02	0.73, 2.31	2.47, 4.91	0.34, 1.61	3.40, 6.15	0.20, 1.32	3.97, 6.89
	0.15	1.76, 3.98	2.43, 4.93	1.03, 2.87	3.50, 6.38	0.77, 2.45	4.10, 7.16	0.36, 1.71	4.30, 7.42
0.6	0.2	2.17, 4.66	3.10, 5.94	1.68, 3.94	3.93, 7.06	1.38, 3.50	4.47, 7.74	0.82, 2.60	4.89, 8.29
	0.25	3.08, 6.03	3.86, 7.08	2.54, 5.28	4.42, 7.82	1.68, 4.04	4.88, 8.40	0.97, 2.93	5.11, 8.69
	0.3	3.78, 7.11	4.51, 8.06	3.19, 6.31	4.99, 8.69	2.15, 4.84	5.12, 8.85	1.26, 3.49	5.73, 9.63
	0.05	0.06, 0.96	0.19, 1.25	0.01, 0.81	0.19, 1.25	0.00, 0.66	0.32, 1.53	0.00, 0.66	0.39, 1.66
	0.1	0.13, 1.17	0.20, 1.32	0.13, 1.17	0.27, 1.47	0.07, 1.02	0.34, 1.61	0.01, 0.86	0.81, 2.45
	0.15	0.21, 1.40	0.29, 1.55	0.14, 1.24	0.36, 1.71	0.07, 1.08	0.95, 2.73	0.01, 0.91	1.86, 4.12
0.7	0.2	0.22, 1.49	0.30, 1.65	0.22, 1.49	1.01, 2.90	0.15, 1.32	1.58, 3.79	0.01, 0.97	2.68, 5.38
	0.25	0.32, 1.76	0.59, 2.27	0.24, 1.58	1.79, 4.20	0.16, 1.40	2.21, 4.82	0.08, 1.22	3.19, 6.18
	0.3	0.54, 2.26	1.70, 4.17	0.54, 2.26	2.38, 5.17	0.35, 1.89	2.84, 5.82	0.17, 1.51	3.78, 7.11
	0.05	0.00, 0.66	0.01, 0.81	0.00, 0.66	0.06, 0.96	0.00, 0.49	0.21, 1.11	0.00, 0.49	0.19, 1.25
	0.1	0.01, 0.86	0.07, 1.02	0.00, 0.69	0.13, 1.17	0.00, 0.69	0.20, 1.32	0.00, 0.52	0.34, 1.61
	0.15	0.07, 1.08	0.14, 1.24	0.00, 0.73	0.21, 1.40	0.00, 0.73	0.36, 1.71	0.00, 0.73	0.44, 1.86
0.8	0.2	0.08, 1.14	0.15, 1.32	0.01, 0.97	0.30, 1.65	0.01, 0.97	0.38, 1.81	0.00, 0.78	0.64, 2.29
	0.25	0.16, 1.40	0.32, 1.76	0.08, 1.22	0.41, 1.93	0.01, 1.03	0.69, 2.44	0.01, 1.03	0.78, 2.61
	0.3	0.26, 1.70	0.35, 1.89	0.26, 1.70	0.64, 2.44	0.09, 1.31	0.74, 2.62	0.01, 1.10	0.84, 2.79

DHISFD3SET is better compared to that of DIFD algorithm.

6.6 Conclusion

In this chapter a robust distributed self diagnosis algorithm for diagnosing the hard and intermittent faulty sensor nodes in WSNs is presented. The inconsistent behavior of the intermittent faulty sensor node is modeled and simulated by using the Bernoulli distribution. The existing DIFD algorithm is compared with the proposed DHISFD3SET algorithm using generic parameters. The simulation result shows that the proposed DHISFD3SET algorithm is improved by 10%, 6%, and 8% over the DIFD algorithm in diagnosis accuracy, false alarm rate, and false positive rate when fault probability is 30%, intermittent fault probability 70% and average degree is

20 and the network size 1024. The algorithm detects the intermittently faulty sensor nodes in all possible faulty scenarios. The modified three sigma edit test based fault diagnosis method reduces the number of iterations required to diagnose the intermittent fault which make the algorithm energy as well as memory efficient.

Chapter 7

Conclusion and Future Scope

Chapter 7

Conclusion and Future Scope

The work in this thesis is based on the statistical approach for fault diagnosis of wireless sensor networks. The overall contributions of the thesis are reported here. Comparison result shows that the proposed algorithms perform better as compared to the existing distributed fault diagnosis algorithms. Future research problems are outlined for extension of this work.

7.1 Conclusion

In this thesis, four distributed self fault diagnosis algorithms have been proposed to diagnose both hard and soft faulty sensor nodes in wireless sensor networks (WSNs). The algorithms are based on a realistic fault model such as stuck at zero, stuck at one, random, hard fault and intermittent fault. All the algorithms are evaluated analytically as well as through simulations using NS3 simulator.

Distributed self fault diagnosis algorithm using neighbor coordination (DSFDNC) has been proposed to minimize the amount of communication overheads present in conventional distributed fault diagnosis algorithm. In neighbor coordination approach each sensor node collects data from the neighboring sensor nodes and estimates its own fault status from neighbor's data. The accuracy and completeness of the DSFDNC algorithm are evaluated. The simulation results show that the diagnosis accuracy, and false positive rate of the new algorithm are improved by 3%, and 1% respectively as compared to that of DFD and IDFD algorithms when the average degree of the network is 15 and network size is 512. The DSFDNC algorithm needs two messages from the neighboring nodes to diagnose the faulty sensor node, unlike 5 and 3 messages for DFD and IDFD algorithms respectively. Hence the algorithm outperforms the DFD and IDFD algorithms by providing higher network lifetime and lower diagnosis latency.

In order to improve the performance of the distributed self fault diagnosis algorithm in sparse WSNs, a distributed self-fault diagnosis algorithm using hypothesis testing (DSFDHT) is proposed based on the neighbor coordination approach. The presence of faulty sensor nodes in the neighbors and the probable fault status of the neighboring sensor nodes are predicted using Newman-Pearson (NP) testing method. From the simulation, it is evident that the diagnosis accuracy, false positive rate, and false alarm rate of DSFDHT algorithm are improved by 2%, 3%, and 2% respectively as compared to the DSFDNC algorithm for the same network configuration. The proposed new algorithm diagnoses the faulty sensor nodes with more than 98% diagnosis accuracy for a wide range of fault probabilities and maintain a negligible (at max 6%) false alarm rate for lower connected network.

A modified three sigma edit test based distributed self fault diagnosis algorithm (DSFD3SET) for large scale WSNs has been proposed. The aim of the algorithm is to reduce the number of message exchanges over the network and enhance the diagnosis accuracy. Each sensor node collects data from the neighbors and predicts the fault status of its own using modified three sigma edit test. The algorithm needs one message exchange among the neighbors to diagnose the fault status, unlike 2 messages in DSFDNC and DSFDHT algorithms. The diagnosis accuracy of the DSFD3SET algorithm is improved by 4%, 6% and 7% as compared to DSFDHT, DFD and IDFD algorithms respectively, when the average degree of the network is 25. The algorithm outperforms over the distributed fault diagnosis algorithms by providing lower false alarm rate, false positive rate and high diagnosis accuracy with lower confidence interval. The algorithm is energy efficient and increases the lifetime of the network.

A distributed self fault diagnosis algorithm to diagnose the intermittent faults in WSN (DSIFD3SET) has been proposed based on the modified three sigma edit test.

The intermittent faulty behavior of the sensor node is modeled using the Bernoulli distribution function. Due to the use of robust statistical test for repeated fault detection, less number of iterations are required to identify the intermittent fault compared to the existing algorithms. The performance of the DSIFD3SET algorithm is compared with the DIFD algorithm in terms of diagnosis accuracy, false alarm rate and false positive rate, which shows that the suggested scheme outperforms over others and diagnoses intermittently faulty sensor nodes in all possible faulty scenarios considered. The confidence interval of diagnosis accuracy and false alarm rate are analyzed and found that the new algorithm performance is better. The proposed DSIFD3SET algorithm also increases the network life time by 33% as compared to the DIFD algorithm. The overall comparison of all the algorithms is tabulated in Table 7.1.

Table 7.1: Comparison of the DSFDNC, DSFDHT, DSFD3SET, and DHISFD3SET algorithms

Criteria	**DSFDNC Algorithm**	**DSFDHT Algorithm**	**DSFD3SET Algorithm**	**DHISFD3SET Algorithm**
Network Topology	Arbitrary network	Arbitrary network	Arbitrary network	Arbitrary network
Network Size	512	512	512	1024
Fault Type	Hard and Soft fault	Soft fault	Hard and soft fault	Hard and intermittent fault
Fault Behavior	Once fault occurs in a sensor node, it persists.	Once fault occurs in a sensor node, it persists.	Once fault occurs in a sensor node, it persists.	Once fault occurs in a sensor node, it persists as intermittent fault.
Data Model	Gaussian distribution with constant mean, different variance.	Gaussian distribution with constant mean, different variance.	Gaussian distribution with constant mean, different variance.	Gaussian distribution with constant mean, different variance.
Testing Mechanism	Neighbor coordination	Hypothesis testing	Three sigma edit test and Modified three sigma edit test	Modified three sigma edit test
Performance Evaluation Parameters	DA,FAR, FPR, DL, EC, ME, NLT	DA,FAR, FPR, DL, EC, ME, NLT	DA,FAR, FPR, DL, EC, ME, NLT	DA,FAR, FPR
Suitability (Network type)	Sparse WSNs (Average degree less than 10)	Sparse WSNs (Average degree less than 10)	Dense WSNs (Average degree more than 10)	Dense WSNs (Average degree more than 10)

DA: Diagnosis Accuracy, FAR: False Alarm Rate, FPR: False Positive Rate
DL: Diagnosis Latency, EC: Total Energy Consumption, ME: Total Message Exchange, NLT: Network Life Time

7.2 Future Scope

The proposed distributed self-fault diagnosis algorithms presented in the thesis are based on the assumption that the network topology is static i.e., the position of a sensor node is not allowed to change during the network life time. In future, the fault diagnosis algorithms is to be developed and evaluated for dynamic topology networks in which sensor nodes join and leave the network during the diagnosis time. The proposed algorithms have also not considered the channel state (faulty or fault free) between sensor nodes. In order to design a robust WSN considering the fault status of both the channel and sensor nodes is our future work. The presented algorithms in Chapters 5 and 6 are mostly suitable for dense WSN. To make those algorithms feasible over sparse WSN, the clustering algorithms (i.e., grouping the large WSNs into a set of clusters) can be used in the proposed algorithms to collect data from multi-hop sensor nodes.

In the proposed distributed self-fault diagnosis algorithms, a hard faulty node is usually isolated from the network. However, in reality, all the individual internal components of a sensor node may not be faulty. The distributed self-fault diagnosis algorithms considering partial failures in a sensor node is to be extended so that it can address a fault model with internal component failures in a sensor node and re-utilization of a sensor node for different works in WSNs. The distributed self-fault diagnosis algorithms are to be investigated to detect the spike and byzantine faulty sensor nodes due to the failure of internal components. In Chapter 6, intermittent fault is considered in which fault state of a sensor node is changed arbitrarily with respect to time. The stochastic behavior of a sensor node is to be modeled using Hidden Markov Model (HMM) to characterize the realistic distributed self-intermittent fault diagnosis algorithm for WSN.

The data model for all the proposed distributed self-fault diagnosis algorithms in the thesis is assumed to be common irrespective of the deployment scenarios. In future, different data models according to the specific deployment scenarios of WSN in the field environment are to be investigated and to be utilized for developing efficient diagnosis algorithms to further enhance the performance of the WSN. In fact, we are in the process of designing and developing a WSN which will serve as a real test bed to evaluate the performance of proposed distributed self-fault diagnosis algorithms as well as fault diagnosis algorithms those are to be investigated in future.

Bibliography

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] I.F.Akyildiz, W. Su, Y.Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Trans. on Commun. Magazine*, vol. 40, pp. 102–114, 2002.

[3] I. Banerjee, P. Chanak, H. Rahaman, and T. Samanta, “Effective fault detection and routing scheme for wireless sensor networks,” *Computers & Electrical Engineering*, vol. 40, no. 2, pp. 291–306, 2014.

[4] S. H. Hosseini, J. G. Kuhl, and S. M. Reddy, “On self-fault diagnosis of the distributed systems,” *IEEE Trans. Computers*, vol. 37, no. 2, pp. 248–251, 1988.

[5] A. Sengupta and A. Dahbura, “On self-diagnosable multiprocessor systems: Diagnosis by the comparison approach,” *IEEE Trans. on computers*, vol. 41, no. 11, pp. 1386–1396, 1992.

[6] J. Chen, S. Kher, and A. Somani, “Distributed fault detection of wireless sensor networks,” in *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, ser. DIWANS ’06, New York, NY, USA, 2006, pp. 65–72.

[7] M. Panda and P. Khilar, “Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test,” *Ad Hoc Networks*, vol. 25, pp. 170 – 184, 2015.

[8] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *IEEE Trans. on Dependable Secure Computing*, vol. 1, pp. 11–33, 2004.

[9] P. Barooah, H. Chenji, R. Stoleru, and T. Kalmar Nagy, “Cut detection in wireless sensor networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 483–490, 2012.

[10] M. Panda and P. M. Khilar, “Distributed soft fault detection algorithm in wireless sensor networks using statistical test,” in *IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC,2012)*, 2012, pp. 195–198.

[11] A. Bondavalli, S. Chiaradonna, F. di Giandomenico, and F. Grandoni, "Threshold-based mechanisms to discriminate transient from intermittent faults," *IEEE Trans. on Computers*, vol. 49, no. 3, pp. 230–245, 2000.

[12] P. Jalote, *Fault tolerance in distributed systems.* Prentice Hall, 1994.

[13] D. P. Siewiorek and R. S. Swmlz, *Reliable Computer System Design and Evaluation.* Digital Press, 1992.

[14] M. Barborak, A. Dahbura, and M. Malek, "The consensus problem in fault-tolerant computing," *ACM Computing Survey*, vol. 25, pp. 171–220, 1993.

[15] M. Elhadef, A. Boukerche, and H. Elkadiki, "A distributed fault identification protocol for wireless and mobile ad hoc networks," *Parallel and Distributed Computing*, vol. 68, no. 3, pp. 321 – 335, 2008.

[16] C. Lo, J. P. Lynch, and M. Liu, "Distributed reference-free fault detection method for autonomous wireless sensor networks," *IEEE Trans. on Sensors Journal*, vol. 13, no. 5, pp. 2009–2019, 2013.

[17] C. Lo, J. Lynch, and M. Liu, "Reference-free detection of spike faults in wireless sensor networks," in *Resilient Control Systems (ISRCS), 2011 4th International Symposium on*, 2011, pp. 148–153.

[18] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Trans. on Sensor Networks (TOSN)*, vol. 6, no. 3, pp. 1–23, 2010.

[19] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. on Sensor Networks*, vol. 5, pp. 1–29, 2009.

[20] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Communication ACM*, vol. 47, pp. 34–40, 2004.

[21] N. Ramanathan, T. Schoellhammer, D. Estrin, M. Hansen, T. Harmon, E. Kohler, and M. Srivastava, "The final frontier: Embedding networked sensors in the soil," UCLA, Center for Embedded Networked Computing, Tech. Rep. CENS-TR-68, 1995.

[22] H. Li, M. Price, J. Stott, and I. Marshall, "The development of a wireless sensor network sensing node utilising adaptive self-diagnostics," in *Self-Organizing Systems*, 2007, pp. 30–43.

[23] E. Elnahrawy and B. Nath, "Cleaning and querying noisy sensors," in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003, pp. 78–87.

[24] N. Ramanathan, L. Balzano, M. Burt, D. Estrin, T. Harmon, C. Harvey, J. Jay, E. Kohler, S. Rothenberg, and M. Srivastava, "Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks," Center for Embedded Networked Sensing, UCLA and Department of Civil and Environmental Engineering, MIT, Tech. Rep., 2006.

[25] M. H. Lee and Y. H. Choi, "Fault detection of wireless sensor networks," *Computer Communications*, vol. 31, no. 14, pp. 3469–3475, 2008.

[26] G. J. Liang, X. Y. Jun, and L. X. Wei, "Weighted-median based distributed fault detection for wireless sensor networks," *Journal of Software*, vol. 18, no. 5, pp. 1208–1217, 2007.

[27] M. Malek, "A comparison connection assignment for diagnosis of multiprocessor systems," in *Proceedings of the 7th annual symposium on Computer Architecture*, 1980, pp. 31–36.

[28] D. M. Blough and H. W. Brown, "The broadcast comparison model for on-line fault diagnosis in multicomputer systems: Theory and implementation," *IEEE Trans. on Computers*, vol. 48, no. 5, pp. 470–493, 1999.

[29] X. Yang, G. M. Megson, and D. J. Evans, "A comparison-based diagnosis algorithm tailored for crossed cube multiprocessor systems," *Microprocessors and Microsystems*, vol. 29, no. 4, pp. 169–175, 2005.

[30] X. Yang and Y. Y. Tang, "Efficient fault identification of diagnosable systems under the comparison model," *IEEE Trans. on computers*, vol. 56, no. 12, pp. 1612–1618, 2007.

[31] S.-Y. Hsieh and Y.-S. Chen, "Strongly diagnosable product networks under the comparison diagnosis model," *IEEE Trans. on Computers*, vol. 57, no. 6, pp. 721–732, 2008.

[32] G. Y. Chang, G. H. Chen, and G. J. Chang, "(t, k) - diagnosis for matching composition networks under the mm* model," *IEEE Trans. on computers*, vol. 56, no. 1, pp. 73–79, 2007.

[33] C. H. Tsai, "A quick pessimistic diagnosis algorithm for hypercube-like multiprocessor systems under the pmc model," *IEEE Trans. on Computers*, vol. 62, no. 2, pp. 259–267, 2013.

[34] G. Y. Chang, "Conditional (t,k)-diagnosis under the pmc model," *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, no. 11, pp. 1797–1803, 2011.

[35] E. P. Duarte, A. Weber, and K. V. O. Fonseca, "Distributed diagnosis of dynamic events in partitionable arbitrary topology networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 230–245, 2012.

[36] S. Chessa and P. Santi, "Comparison-based system-level fault diagnosis in ad hoc networks," in *20th IEEE Symposium on Reliable Distributed Systems, 2001*, 2001, pp. 257–266.

[37] B. Krishnamachari and S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Trans. on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[38] T. N. Simulator, http://nsnam.org.

[39] M. Panda and P. M. Khilar, "Energy efficient soft fault detection algorithm in wireless sensor networks," in *IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC,2012)*, 2012, pp. 801–805.

[40] P. Jiang, "A new method for node fault detection in wireless sensor networks," *Journals on sensors*, vol. 9, no. 2, pp. 1282–1294, 2009.

[41] F. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Trans. on Communication Magazine*, vol. EC-16, no. 6, pp. 848–854, 1967.

[42] M. Panda and P. M. Khilar, "An efficient fault detection algorithm in wireless sensor network," in *International Conference on Contemporary Computing (IC3 2011)*, 2011, pp. 279–288.

[43] M. Nandi, A. Dewanji, B. K. Roy, and S. Sarkar, "Model selection approach for distributed fault detection in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, no. 48234, pp. 1–12, 2014.

[44] K. Liu, Q. Ma, X. Zhao, and Y. Liu, "Self-diagnosis for large scale wireless sensor networks," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1539–1547.

[45] M. Elhadef, A. Boukerche, and H. Elkadiki, "Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols," in *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, ser. MobiWac '06, New York, NY, USA, 2006, pp. 18–27.

[46] M. Malek, "A comparison connection assignment for diagnosis of multiprocessor systems," in *Proceedings of the 7th annual symposium on Computer Architecture*, ser. ISCA '80, New York, NY, USA, 1980, pp. 31–36.

[47] K.-Y. Chwa and S. L. Hakimi, "Schemes for fault tolerant computing: A comparison of modularly redundant and t-diagnosable systems," *Information and Control*, vol. 49, no. 3, pp. 212 – 238, 1981.

[48] H. C. Shih, J. H. Ho, B. Y. Liao, and J. S. Pan, "Fault node recovery algorithm for a wireless sensor network," *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2683–2689, 2013.

[49] S. Kreutzer and S. Hakimi, "Adaptive fault identification in two new diagnostic models," in *Proc. 21st Allerton Conf. Comm., Control and Computing*, 1983, pp. 353–362.

[50] G. G. L. Meyer, "A diagnosis algorithm for the bgm system level fault model," *IEEE Trans. on Computers*, vol. C-33, no. 8, pp. 756–758, 1984.

[51] D. Blough and H. Brown, "The broadcast comparison model for on-line fault diagnosis in multicomputer systems: theory and implementation," *IEEE Trans. on Computers*, vol. 48, no. 5, pp. 470–493, 1999.

[52] E. Mourad and A. Nayak, "Comparison-based system-level fault diagnosis: A neural network approach," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1047–1059, 2012.

[53] L. C. P. Albini, E. P. Duarte, and R. P. Ziwich, "A generalized model for distributed comparison-based system-level diagnosis," *Springer Journal of the Brazilian Computer Society*, vol. 10, no. 3, pp. 42–54, 2004.

[54] C. Hsin and M. Liu, "Self-monitoring of wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 462–476, 2006.

[55] G. J. Liang, X. Y. Jun, and L. X. Wei, "Weighted median based distributed fault detection for wireless sensor networks," *Journal of Software*, vol. 18, no. 5, pp. 1208–1217, 2007.

[56] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 2005, pp. 902–913.

[57] S. Ji, S. f. Yuan, T. M. huai, and C. Tan, "Distributed fault detection for wireless sensor based on weighted average," in *Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2010, pp. 57–60.

[58] M. Nandi, A. Nayak, B. Roy, and S. Sarkar, "Hypothesis testing and decision theoretic approach for fault detection in wireless sensor networks," *arXiv preprint arXiv:1203.0541*, 2012.

[59] B. C. Lau, E. W. Maa, and T. W. Chow, "Probabilistic fault detector for wireless sensor network," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3703–3711, 2014.

[60] J. Z. He, Z. H. Zhou, X. R. Yin, and S. F. Chen, "Using neural networks for fault diagnosis," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 5, 2000, pp. 217–220.

[61] M. Elhadef and A. Nayak, "Efficient symmetric comparison-based self-diagnosis using backpropagation artificial neural networks," in *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*, 2009, pp. 264–271.

[62] A. Mohaptra and P. Khilar, "Detection and diagnosis of node failure in wireless sensor networks: A multiobjective optimization approach," *Swarm and Evolutionary Computation*, vol. 13, pp. 74 – 84, 2013.

[63] M. Elhadef and B. Ayeb, "Efficient comparison-based fault diagnosis of multiprocessor systems using an evolutionary approach," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, 2001, pp. 1–6.

[64] M. Elhadef and A. Nayak, "Efficient symmetric comparison-based self-diagnosis using backpropagation artificial neural networks," in *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*, 2009, pp. 264–271.

[65] S. Yuan and F. Chu, "Fault diagnosis based on support vector machines with parameter optimisation by artificial immunisation algorithm," *Mechanical Systems and Signal Processing*, vol. 21, no. 3, pp. 1318–1330, 2007.

[66] Z. Ji, W. Bing-shu, M. Yong-guang, Z. Rong-hua, and D. Jian, "Fault diagnosis of sensor network using information fusion defined on different reference sets," in *International Conference on Radar*, 2006, pp. 1–5.

[67] A. Jabbari, R. Jedermann, and W. Lang, "Application of computational intelligence for sensor fault detection and isolation," in *World Academy of Science, Engineering and Technology*, 2007, pp. 265–270.

[68] A. Moustapha and R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Trans. on Instrumentation and Measurement*, vol. 57, no. 5, pp. 981–988, 2008.

[69] J. Barron, A. Moustapha, and R. Selmic, "Real-time implementation of fault detection in wireless sensor networks using neural networks," in *Fifth International Conference on Information Technology: New Generations*, 2008, pp. 378–383.

[70] S. Guo and Z. Zhong, "FIND: faulty node detection for wireless sensor networks," in *SenSys'09*, 2009, pp. 1–14.

[71] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, ser. SenSys '05, New York, NY, USA, 2005, pp. 255–267.

[72] A. K. Somani, "Sequential fault occurrence and reconfiguration in system level diagnosis," *IEEE Trans. on Computers*, vol. 39, no. 12, pp. 1472–1475, 1990.

[73] E. P. Duarte Jr and T. Nanya, "A hierarchical adaptive distributed system-level diagnosis algorithm," *IEEE Trans. on Computers*, vol. 47, no. 1, pp. 34–45, 1998.

[74] R. P. Bianchini Jr and R. W. Buskens, "Implementation of online distributed system-level diagnosis theory," *IEEE Trans. on Computers*, vol. 41, no. 5, pp. 616–626, 1992.

[75] S. Chessa and P. Santi, "Crash faults identification in wireless sensor networks," *Computer Communications*, vol. 25, no. 14, pp. 1237–1282, 2002.

[76] X. Xu, W. Chen, J. Wan, and R. Yu, "Distributed fault diagnosis of wireless sensor networks," in *Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on*, 2008, pp. 148–151.

[77] T. Saha and S. Mahapatra, "Distributed fault diagnosis in wireless sensor networks," in *Process Automation, Control and Computing (PACC), 2011 International Conference on*, 2011, pp. 1–5.

[78] T. Y. Wang, L. Y. Chang, D. R. Duh, and J. Y. Wu, "Distributed fault-tolerant detection via sensor fault detection in sensor networks," in *10th International Conference on Information Fusion, 2007*, 2007, pp. 1–6.

[79] A. Haeberlen, P. Kouznetsov, and P. Druschel, "The case for byzantine fault detection," in *Proceedings of the 2nd conference on Hot Topics in System Dependability*, 2006, pp. 1–6.

[80] X. Luo, M. Dong, and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," *IEEE Trans. on Computers*, vol. 55, no. 1, pp. 58–70, 2006.

[81] M. Yu, H. Mokhtar, and M. Merabti, "Fault management in wireless sensor networks," *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 13–19, 2007.

[82] S. Harte, A. Rahman, and K. Razeeb, "Fault tolerance in sensor networks using self-diagnosing sensor nodes," in *The IEE International Workshop on Intelligent Environments, 2005*, 2005, pp. 7–12.

[83] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentell, "Fault tolerance techniques for wireless ad hoc sensor networks," in *Proceedings of IEEE Sensors*, vol. 2, 2002, pp. 1491–1496.

[84] D. Rakhmatov and S. Vrudhula, "Time-to-failure estimation for batteries in portable electronic systems," in *International Symposium on Low Power Electronics and Design*, 2001, pp. 88–91.

[85] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "A discrete-time battery model for high-level power estimation," in *Proceedings of the conference on Design, automation and test in Europe*, 2000, pp. 35–41.

[86] A. Mahapatro and P. M. Khilar, "Online distributed fault diagnosis in wireless sensor networks," *Wireless Personal Communication*, vol. 71, no. 3, pp. 1931–1960, 2013.

[87] D. M. Blough, G. F. Sullivan, and G. M. Masson, "Intermittent fault diagnosis in multiprocessor systems," *IEEE Trans. on computers*, vol. 41, no. 11, pp. 1430–1441, 1992.

[88] P. M. Khilar and S. Mahapatra, "Intermittent fault diagnosis in wireless sensor networks," in *ICIT'07 Proceedings of the 10th International Conference on Information Technology*. IEEE Computer Society, Washington DC, USA, 2007, pp. 145–147.

[89] S.-J. Yim and Y.-H. Choi, "An adaptive fault-tolerant event detection scheme for wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 2332–2347, 2010.

[90] K.-F. Ssu, C.-H. Chou, H. C. Jiau, and W.-T. Hu, "Detection and diagnosis of data inconsistency failures in wireless sensor networks," *Computer Networking*, vol. 50, pp. 1247–1260, 2006.

[91] T. J. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Palo Alto: MIT Press and McGraw-Hill, 1990.

[92] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in wireless ad hoc sensor networks," *IEEE Trans. on computers*, vol. 52, no. 6, pp. 753–763, 2003.

[93] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, ser. HICSS '00, 2000, pp. 8020–8030.

[94] Y. J. Zhao, R. Govindan, and D. Estrin, "Residual energy scan for monitoring sensor networks," in *IEEE WCNC,02*, 2002, pp. 256–262.

[95] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, 1st ed. PHI Publication, 1993.

[96] A. Somani, V. Agarwal, and D. Avis, "A generalized theory for system level diagnosis," *IEEE Trans. on Computers*, vol. C-36, no. 5, pp. 538–546, 1987.

[97] A. Somani and V. Agarwal, "Distributed diagnosis algorithms for regular interconnected structures," *IEEE Trans. on Computers*, vol. 41, no. 7, pp. 899–906, 1992.

[98] MATLAB, http://www.mathworks.in/, 2014.

[99] T. Clouqueur, K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," *IEEE Trans. on Computers*, vol. 53, no. 3, pp. 320–333, 2004.

[100] B. Krishnamachari and S. Iyenger, "Distributed bayesian algorithm for fault tolerant event region detection in wireless sensor networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 24, no. 8, pp. 1525–1534, 2004.

[101] T. Pham-Gia and T. L. Hung, "The mean and median absolute deviations," *Mathematical and Computer Modeling*, vol. 34, pp. 921–936, 2001.

[102] P. J. Rousseeuw and C. Croux, "The mean and median absolute deviations," *Journal of the American Statistical Association*, vol. 88, no. 424, Theory and Methods, pp. 1273–1283, 1993.

[103] H. T. FRIIS, "A note on a simple transmission formula," in *Proceedings of the I.R.E. and Waves and Electrons*, 1946, pp. 254–256.

[104] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[105] S. Bavishi and E. Chong, "Automated fault diagnosis using a discrete event systems framework," in *Intelligent Control, 1994., Proceedings of the 1994 IEEE International Symposium on*, 1994, pp. 213–218.

[106] O. Contant, S. Lafortune, and D. Teneketzis, "Diagnosis of intermittent failures," *Discrete event dynamic systems: Theory and applications.*, vol. 14, no. 2, pp. 171–202, 2004.

[107] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

[108] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, no. 1, pp. 97–130, 1987.

[109] S. Jiang, R. Kumar, and H. E. Garcia, "Diagnosis of repeated/intermittent failures in discrete event systems," *IEEE Trans. On Robotics and Automation*, vol. 19, no. 2, pp. 310–323, 2003.

[110] S. Jiang and R. Kumar, "Diagnosis of repeated failures for discrete event systems with linear-time temporal-logic specifications," *IEEE Trans. on Automation Science and Engineering*, vol. 3, no. 1, pp. 47–59, 2006.

[111] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Proc. Discrete Event Dynamical Syst.: Theory Appl.*, vol. 10, pp. 33–79, 2000.

[112] A. Mahapatro and A. K. Panda, "Choice of detection parameters on fault detection in wireless sensor networks: A multiobjective optimization approach," *Wireless Personal Communication*, vol. 78, no. 1, pp. 649–669, 2014.

Dissemination

Journals

[1] Meenakshi Panda, P. M. Khilar, "Distributed Self Fault Diagnosis Algorithm for Large Scale Wireless Sensor Networks using Modified Three Sigma Edit Test," *Ad Hoc Networks, Elsevier*, 25(2015), 170-184, 2014 .

[2] Meenakshi Panda, P. M. Khilar, "Energy Efficient Distributed Fault Identification Algorithm in Wireless Sensor Networks," *Journal of Computer Networks and Communications, Hindawi Publisher*, Volume 2014, Article ID 323754, 1-16, 2014.

[3] Meenakshi Panda, P. M. Khilar, "Distributed Byzantine Fault Detection Technique in Wireless Sensor Networks Based on Hypothesis Testing," *Computers and Electrical Engineering, Elsevier*, **Available Online** DOI-http://dx.doi.org/10.1016/j.compeleceng.2015.06.024.

[4] Meenakshi Panda, P. M. Khilar, "Robust Distributed Intermittent Fault Diagnosis in Large Scale Wireless Sensor Networks," *Ad Hoc Networks, Elsevier*, **First Revision submitted to the Journal**.

Conferences

[1] Meenakshi Panda, P. M. Khilar, "Energy Efficient Soft Fault Detection Algorithm in Wireless Sensor Networks," in *IEEE International Conference on Parallel, Distributed and Grid Computing(PDGC,2012)*, at JP University of information and technology, Waknaghat, Solan, India, pp. 801–805, Dec. 2012.

[2] Meenakshi Panda, P. M. Khilar, "Distributed Soft Fault Detection Algorithm in Wireless Sensor Networks using Statistical Test," in *IEEE International Conference on Parallel, Distributed and Grid Computing(PDGC,2012)*, at JP University of information and technology, Waknaghat, Solan, India, pp. 195–198, Dec. 2012.

[3] Meenakshi Panda, P. M. Khilar, "Distributed Fault Tolerant Estimation in Wireless Sensor Network Using Robust Diffusion Adaptation," in *8th International Conference on Distributed Computing and Internet Technology, ICDCIT-2012*, at KIIT University, Bhubaneswar, India, Feb. 2012, pp. 259–260, LNCS Springer.

[4] Meenakshi Panda, P. M. Khilar, "An Efficient Fault Detection Algorithm in Wireless Sensor Network," in *International Conference on Contemporary Computing (IC3 2011)*, at JP Institute of Information Technology, Noida, India, pp. 279–288, Aug. 2011, CCIS Springer.

BIO-DATA

Meenakshi Panda

Date of Birth: 30^{th} June, 1982

Correspondence:
PhD Scholar, Department of Computer Science and Engineering,
National Institute of Technology Rourkela, India – 769 008.

Ph: +91 94397 24318 (M)

e-mail: 509cs106@nitrkl.ac.in, meenakshi.panda1@gmail.com

Qualification

- Ph.D. (Continuing)
  National Institute of Technology Rourkela, Odisha, India
- M.Tech. (Computer Science and Engineering)
  Utkal University, Odisha, India [First division]
- M.Sc. (Computer Science)
  Utkal University, Bhubaneswar, Odisha, India [First class with distinction]
- B.Sc. (Computer Science)
  Utkal University, Bhubaneswar, Odisha, India [First class with distinction]
- +2 (Science)
  Council of Higher Secondary Education, Odisha, India [First division]
- 10th
  Board of Secondary Education, Odisha, India [First division]

Professional Experience

Assistant Professor, Silicon Institute of Technology, Bhubaneswar, Odisha, India, January 2006 – December 2009

Publications

- 03 Journal Articles
- 06 Conference Articles